**1.Introduction**

With the rapid development of information technology, commenting on the Internet has become an important way for people to express their views and pass on their experiences. However, the explosive growth of information makes it more difficult to obtain useful information from them. Sentiment analysis, which automatically captures the opinions expressed in comment texts, facilitates users' access to opinion information by calculating the opinions, emotions, evaluations and attitudes in the texts and automating sentiment recognition.

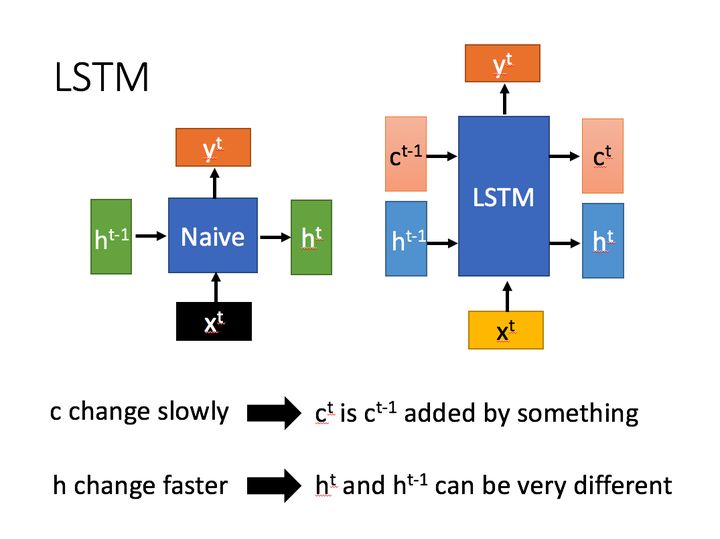
Text sentiment analysis can be divided into chapter-level sentiment analysis, sentence-level sentiment analysis and aspect-level sentiment analysis. In this project, I will apply two methods on sentence-level sentiment analysis and aspect-level sentiment analysis respectively, in order to compare both methods and problems and have a better understanding of the whole sentiment analysis task.

**2.Theory**

2.1 LSTM

Long short-term memory (LSTM) is a special kind of RNN, which is designed to solve the gradient disappearance and gradient explosion problems during the training of long sequences. LSTM can perform better on longer sequences than normal RNNs.

The main input and output differences between the LSTM structure (figure right) and a normal RNN are shown below.



2.2 Attention

Attention is mainly used to solve the problem that it is difficult to obtain a final reasonable vector representation when the input sequence of LSTM/RNN models is long. The attention mechanism can be divided into three steps: first, the input of information; second, the calculation of the attention distribution α; and third, the calculation of a weighted average of the input information based on the attention distribution α.

2.3 BERT

In its paper BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, the BERT model is proposed by Google in 2018 and has become one of the most groundbreaking technologies in NLP in recent years. It has set previous records on 11 NLP domain tasks, such as GLUE, SquAD1.1, MultiNLI, etc.

The [CLS] flag is placed at the beginning of the first sentence and the representation vector C obtained by BERT can be used for subsequent classification tasks.

The [SEP] flag is used to separate two input sentences, e.g. input sentences A and B. The [SEP] flag is added after sentences A, B.

The input Embedding of BERT is obtained by summing three components: Token Embedding, Segment Embedding, and Position Embedding.Token Embedding is the embedding of words, e.g. [CLS] dog, etc., learned by training.Segment Embedding is used to distinguish whether each word belongs to sentence A or sentence B. BERT's Position Embedding is also learned, and the maximum sentence length is assumed to be 512.

**3.Data**

For aspect-level sentiment analysis, I choose semeval-2014 Task 4 dataset. I extract the text and aspect term from Restaurant\_Train.xml and split it into trainset and valset/testset with the ratio of 8 : 2.

For sentence-level sentiment analysis, I choose the IMDB dataset from Kaggle. Since the dataset is so huge, I choose 4000 pieces of text as trainset and 1000 pieces of text as testset.

**4.Method**

For aspect-level sentiment analysis, I choose atae-lstm model. It is based on Attention-based LSTM for Aspect-level Sentiment Classification. In this paper, researchers combine attention with LSTM to solve the aspect level sentiment analysis problem by using attention to obtain contextual information that is more important to different aspects, and achieve good results on the experimental dataset. For the atae-lstm model’s structure, I mainly refer to <https://github.com/songyouwei/ABSA-PyTorch/tree/9a7c6c1c993ad772d5f439d32af4899b6698ea67>.

For sentence-level sentiment analysis, I choose BertForSequenceClassification model. BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train unlabelled text to obtain a deep bi-directional representation through the union of left and right contexts. Thus, with only one additional output layer, the pre-trained BERT model can be fine-tuned to create state-of-the-art results for various NLP tasks.

**5.Results**

5.1 Speed

I use the actual time to determine the speed of model. This can basically show the complexity of models, though choice of optimizer and so on can affect the time.

It takes about 400 seconds for the atae-lstm model to run 10 epoches. While the bert model can only run 1 epoch using the same time.

5.2 Train loss and Accuracy

Accuracy is the basic evaluation metric.

After 10 epoch, the atae-lstm’s train accuacy rises from 0.57 to 0.72, test accuacy is 0.696.

After 5 epocch, the bert model’s train accuacy rises from 0.496 to 0.556, test accuacy is 0.582.

5.3 F1

The F1-score is a metric that takes into account both precision and recall: 2\*precision\*recall / (precision + recall). Since the dataset is picked randomly, in case we have imbalance between positive and negative examples, I use the F1 score.

After 10 epoch, the atae-lstm’s test data F1-score gradually rises from 0.35 to about 0.6.

After 5 epoch, the bert’s test data F1-score is 0.547.

**6.Discussion**

For aspect-level sentiment analysis, We can see that the outcome is gradually good. The accuracy increase steadily and the speed is fast.

However, the outcome for the sentence-level sentiment analysis is not very ideal, but still, the accuracy has a small increase.

The similarity between these two tasks is that the amount of data is similar. But the complexity of the data is different. For task 1, maximum sequence length is 69 while more than 50% of the IMDB text length is higher than 171. In order to reduce the length of the word vector in task 2, I set the max length 64, which may cause the model cannot predict the sentiment more accurately.

Moreover, the bert model is very deep. It has 12 layers.(Although it is not wide. The middle layer is only 1024 compared to 2048 in the previous Transformer model, which confirm the computer image processing idea that deep and narrow is better than shallow and wide.) So it is very time consuming to load or train the bert model. That explains the low speed of bert model.

Next, I want to discuss about why these two model works.

For atae-lstm, word respresentation and aspect embedding are concatenated and fed into the lstm to obtain the hidden vector, then the hidden vector is concatenated with aspect embedding and an attention is done, and the result is output to the fully concatenated layer to give the final prediction. Though the model is simple, it make use of the context information very well.

In a lot of ranks, we can see that bert based model is better than lstm. In terms of the network structure and the final experimental results, the main reasons why bert works better than lstm are focused on the following. Firstly, the lstm is much less capable of extracting features than the Transformer, and secondly, the training data and model parameters for bert are very large, as BERT-base was trained using the BookCorpus dataset of about 800 million words.

However, the lstm and other general models use a language modelling task to obtain an embedding representation of the words in a sentence, which can be applied in any situation. Whereas if we were to use the parameters of BERT, we would have to transform the model into a BERT model in order to use the pre-trained parameters of the BERT model. So bert, although it has great performance, does not fit into any situation seamlessly.

**7. Conclusion**

In conclusion, both methods use the attention. There is actually a paper named “Attention is all you need”. Using attention help us have fewer parameters, fast model and good results. And it is now widely used in NLP problems.

However, someone just point out that the position information lost by attention is actually very important in NLP, and the inclusion of position embedding in the feature vector in is only a stop-gap measure. So we may need to do more work to use the position information when processing NLP problems.

We can also learn that pre-training is important. As the researchers of bert model said, "We believe that this is the first work to demonstrate that scaling to extreme model sizes also leads to large improvements on very small-scale tasks, provided that the model has been sufficiently pre-trained". A good pre-trained model is more likely to succeed.