Attention-Encoded LSTM for Aspect based Sentiment Analysis with RoBERTa

Zhengxu Hou, 3111014920
Fang Xu, 7273293505

University of Southern California
zhengxuh@usc.edu **fangxu@usc.edu**

Abstract

Aspect-Based Sentiment Analysis (ABSA) is becoming popular among industry, which has many applications e.g. in e-commerce, where data and insights from reviews can be leveraged to create value for businesses and customers. Most of previous model contains RNN. And also, it becomes a trend that research want to do parallel network aimed to capture more information. However, RNN is time consuming and cannot acquire the relationship between word and word. To address these issues, this paper proposes an Attention-Encoded LSTM, which is Multi Head Self Attention based so that it will eschews the relationship between context and target. More importantly, our model is small and quick than most model. And for the ABSA task of SemEval 2014 Task 4, our model is better than the baseline of paperwork AEN-Bert. Experiment and analysis demonstrate the effectiveness of our model.

1 Introduction

Sentiment analysis is a vital application and an ever-emerging research area in natural language processing (NLP). The big difference between sentiment analysis and aspect-based sentiment analysis is that the former only detects the sentiment of an overall text, while the latter analyzes each text to identify various aspects and determine the corresponding sentiment for each one.

In other words, instead of classifying the overall sentiment of a text into positive or negative, aspect-based analysis allows us to associate specific sentiments with different aspects of a product or service. The results are more detailed, interesting and accurate because aspect-based analysis looks more closely at the information behind text. Scientists, for example, analyze cells under a microscope so that they can better visualize their components, and aspect-based sentiment analysis follows this principle. when we talk about aspects, we mean the attributes or components of a product or service e.g. ‘the user experience of a new product’, ‘the response time for a query or complaint’ or ‘the ease of integration of new software’.

There are 4 general problems in sentiment analysis:

Irony and sarcasm. Sarcasm means using the same word but expressing the opposite meaning. This easily makes a simple sentiment analysis model confused if the model is not designed deliberately for sarcasm. Sarcasm is very common in user-generated content (UGC), such as product reviews, tweets. Sarcasm detection in sentiment analysis depends highly on the context and environment. Automatic sarcasm detection includes rule-based, statistical, machine learning algorithms and deep learning. Deep learning approach is becoming popular.

Types of negations. Negation can reverse the polarity of words, phrases or sentences. It is important to determine the effective scope of a negation word. The simplest and most popular approach to deal with negations is to mark the words from its occurrence to the next punctuation token.

Word ambiguity. This stop us determining the polarity of an ambiguous word because its meaning depends on the context. It is a challenge to lexicon-based sentiment analysis approaches.

Multipolarity. When a sentence contains multiple aspects and polarities, considering the whole polarity only is misleading. It is where aspect-based sentiment analysis can take advantages.

2 Related Work and background

2.1 ABSA (Aspect Based Sentiment Analysis)

For now, the research approach of the targeted sentiment classification task including non-BERT and BERT methods.

For non-BERT methods, including traditional machine learning methods and neural network methods. For neural network, most research like to use RNN combined with MHA. In 2018, Fan and Feifan propose a Multi-Grained Attention network, and they design an aspect alignment loss to depict the aspect-level interactions among the aspects that have the same context.[1] And for RNN, Peng Chen propose a Recurrent Attention Network, where RNN play an important role in strengthening the expressive the model.[2]

For BERT methods, researchers always put BERT as an embedding layer. In 2019，Zeng and Yang propose a LCF-BERT network, which mainly use MHA network and also use CDM/CDW to capture more relative information.

2.2 BERT (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) has been the state-of-art model for language model pre-training. Its architecture is a multi-layer bidirectional Transformer encoder. One of its distinguished features is that it is capable for various NLP tasks with minimal modifications. When pre-training BERT, it includes a masked LM task and a Next Sentence Prediction (NSP) task. At masked LM task, 15% of tokens are masked; when a token is selected, it has 80% to be changed to [MASK], 10% to be a random token and 10% to be not changed. The NSP task is binarized, i.e., input is a pair of sentences to predict whether the latter is the actual next sentence that follows the former.[3]

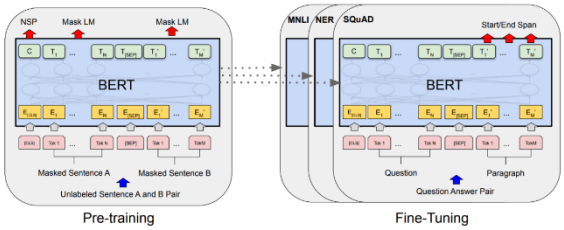

Figure 1 Overall training and fine-tuning procedures of BERT

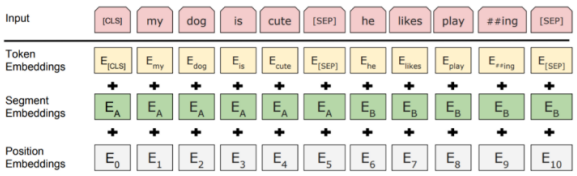

Figure 2 BERT input representation

2.3 Baseline Model (AEN-BERT)

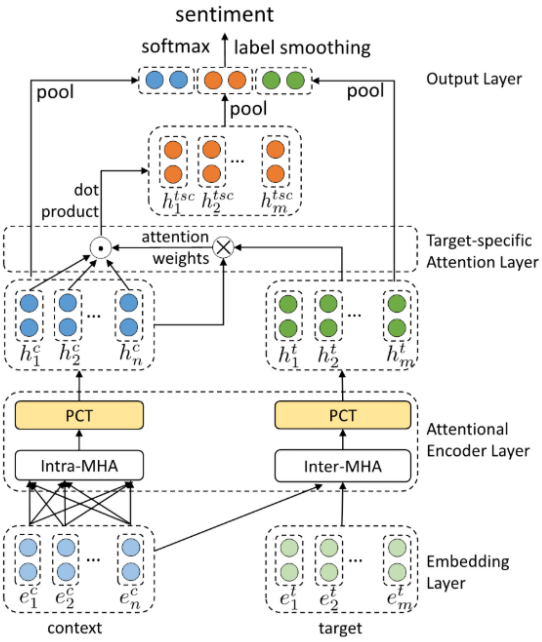

Figure 3 Overall architecture of AEN

Figure 3 illustrates the overall architecture of Attentional Encoder Network (AEN), which mainly consists of an embedding layer, an attentional encoder layer, a target-specific attention layer, and an output layer. During pooling part, it is easy to find that author want to concatenate three vector which come from three different places. For one, through a intranet -MHA and PCT layer, it will output a relationship between each word in this sentence. For the second one, by using an inter-MHA and PCT layer, it will output a relationship between aspect and context. Finally, the third will combine the outputs of previous two procedure to form a new MHA layer.[4]

Our baseline model, which is proposed by Youwei and Jiahai, wants to address the issue of time and memory consuming of RNN network. They propose a novel network which could capture the inter information between context and aspects. But during experiment, we find that actually only

one third of this network could contribute to the scores, and even this good performance also highly relies on BERT embedding layer. And also, this network failed to capture context information in one sentence. For example, if however, but or not exist in this context, this network will only capture wrong features. Although there are one third of this model is about context, but the MHA could only help them capture information in word-word level, not in a semantic or sentence level.

3 Proposed Methodology

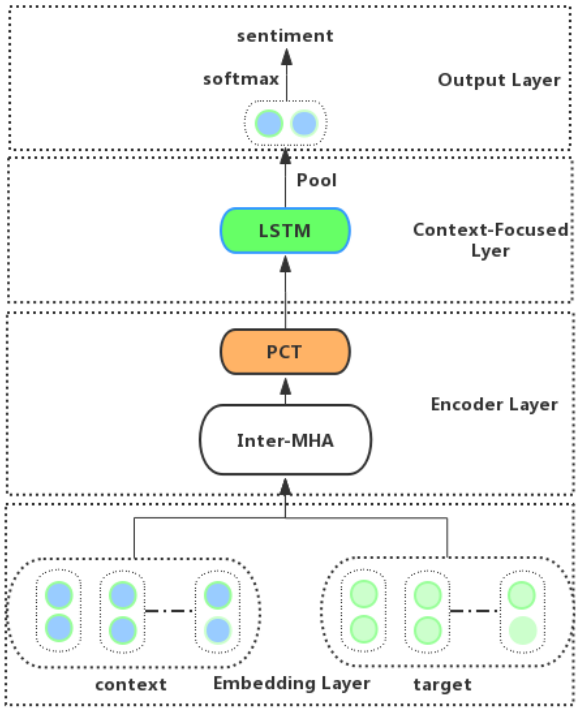

Figure 4 Overall Architecture of proposed AE-LSTM

Overall, the structure has four layers, the first one is embedding layer, which will have a dimension of (768 * hidden) of each word. And then we use inter MHA to compute the how much each word in context and targets are related. And then we use PCT layer. Then, we use LSTM and CDM to let this model focus more on the context. At last, we use mean pooling and softmax to get the results. Final results have three values, one is negative value, one is natural value and the last one is for positive value.

3.1 Roberta Embedding Layer

RoBERTa builds on BERT's language masking strategy and modifies key hyperparameters in BERT, including removing BERT's next-sentence pretraining objective, and training with much larger mini-batches and learning rates. RoBERTa was also trained on an order of magnitude more data than BERT, for a longer amount of time. This allows RoBERTa representations to generalize even better to downstream tasks compared to BERT.[5]

Roberta embedding layer uses pretrained Roberta to get the word embedding. By using the source code of transformer, I could transform given sentence into "[CLS]+ sentence +[SEP]". And then we could compute in next step.

3.2 Attention Encoding Layer

1. Self Attention

The first step is to calculate the Query, Key, and Value matrices. We do that by packing our embeddings into a matrix X, and multiplying it by the weight matrices we've trained (W_Q, W_K, W_V). And then we could get three matrices, which is (Q, K, V). And for details, each matrix represents something, for Query and Key, they could represent the relationship between different words, and for values, this one is actually a weighted number, to weight the final output. And the computation for final matrix is as below:

$$Z = \text{softmax}(\frac{Q \times K^T}{\sqrt{d_k}}) \times V \quad (1)$$

2.Multi Head Self Attention

The major difference between multi head and self-attention is that multi head repeat the process of self-attention. Instead of one group of (Q, K, V), there are several groups of (Q, K, V) matrices. It gives the attention layer multiple "representation subspaces". Each of these sets is randomly initialized. Then, after training, each set is used to project the input embeddings into a different representation subspace. Below is a example plot of 8 heads.

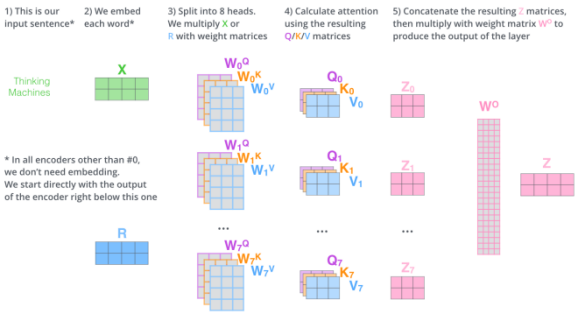

Figure 5 Mechanism of Attention

3.3 PCT (Position-Wise Convolution Trans formation)

This layer is aimed to transform information gathered by MHA. Point-wise means that the kernel size is set to 1. Basically, given a input sequence h, PCT is defined as:

$$PCT(h) = relu(h * W_{pc}^{1} + b_{pc}^{1}) * W_{pc}^{2} + b_{pc}^{2}$$

W is weight matrix and b is bias vector. And in this way, we will get an output of PCT layer as following, the input is $\mathrm{MHA}_{\mathrm{inter}}$.

$$h^{t} = PCT(t^{inter}) \quad (2)$$

3.4 LSTM

In this part, we will describe a long short-term memory (LSTM) model for target-dependent sentiment classification. The reason why we use this part is we need to capture more information from context, where the MHA could only capture information in a word level.

LSTMs are explicitly designed to avoid the long-term dependency problem. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

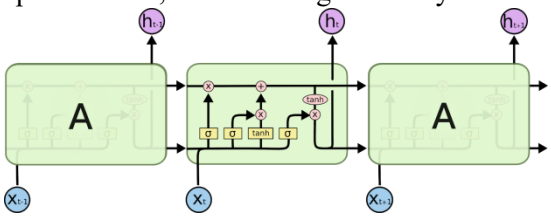

Figure 6 LSTM unit structure

We use LSTM to compute the vector of a sentence from the vectors of words it contains, an illustration of the model is shown in Figure 6. LSTM is a kind of recurrent neural network (RNN), which is capable of mapping vectors of words with variable length to a fixed-length vector by recursively transforming current word vector with the output vector of the previous step h(t-1).

3.5 Output layer:

The final result is from mean pooling, concatenate the output of LSTM as the final comprehensive representation, and then use a fully connected layer to project the concatenated vector into the space of the target C class. And the out is the predicted sentiment.

$$x = W^{T} \times h_{LSTM} + b \quad (3)$$

$$y = soft\max(x) \quad (4)$$

4 Experiments

4.1 Tasks and dataset

In this section, we present the performance of our model on the 3 datasets: SemEval 2014 Task 4 dataset, including Restaurant reviews and Laptop reviews, and ACL 14 Twitter dataset collected by Dong et al..[6][7]

4.2 Experimental settings

We use the default hyperparameters from AEN-BERT before fine-tuning, including dimension of hidden states to 300, Glorot initialization for model weights, training label smoothing parameter to 0.2, L-2 regularization coefficient to 1e-5 and dropout rate to 0.1.[4] The RoBERTa module is pretrained from Transformers v2.2, and the embedding dimension is 768.[8] The LSTM hidden dimension is 300. Adam optimizer is used to train the model. We use accuracy and macro-F1 metrics to evaluate the performance of models.

Table 1 Statistics of datasets

Dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Twitter	1561	173	3127	346	1560	173
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128

Table 2 Main results. The results of AEN are retrieved from the paper.

	Models	Twitter		Restaurant		Laptop	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
AEN	AEN-Glove	0.7283	0.6981	0.8098	0.7214	0.7351	0.6904
	BERT-SPC	0.7355	0.7214	0.8446	0.7698	0.7899	0.7503
	AEN-BERT	0.7471	0.7313	0.8312	0.7376	0.7993	0.7631
Ours	AE-LSTM	0.7587	0.7420	0.8268	0.7415	0.8009	0.7674

4.3 Results and comparisons

Table 2 shows the performance comparison of AE-LSTM with different settings and parameters. We can see that although our model is much simpler than AEN-BERT, it can still perform same or better than AEN-BERT. In detail, AE-LSTM performs better on dataset Twitter, with both more than 1% improvement on accuracy and macro-F1 than AEN-BERT. On dataset Restaurant, AE-LSTM does not outperform BERT-SPC, but is still better than AEN-Glove. This can be understood as the power of the pretrained RoBERTa. On dataset Laptop, AE-LSTM performs slightly better than AEN-BERT.

Table 4 Tuning number of attention heads

# of heads	Laptop	
	Accuracy	Macro-F1
1	0.7947	0.7545
2	0.7806	0.7408
4	0.8009	0.7674
8 (default)	0.7915	0.7549
16	0.7712	0.7216
24	0.7962	0.7605

4.4 Analysis

In this section we conduct analysis to our model with results of fine-tuning.

Table 3 shows the performance of AE-LSTM on different numbers of attention heads. This hyper-parameter controls the fineness of the MHA module. We can see that setting this value too high or too low are both harmful.

Table 4 shows the performance of AE-LSTM on different values of LSTM hidden dimension. The tremendous number of parameters in LSTM cannot give much help to our model.

Table 3 Tuning LSTM hidden dimension

LSTM hid_dim	Laptop	
	Accuracy	Macro-F1
300 (default)	0.7915	0.7549
600	0.7915	0.7583
900	0.7900	0.7530

Table 5 shows the performance of AE-LSTM on different values of LSTM layers. Similar with LSTM hidden dimension, a multi-layer LSTM with multiplied amounts of parameters cannot give much help to our model.

Table 5 Tuning LSTM layers

LSTM layers	Laptop	
	Accuracy	Macro-F1
1 (default)	0.7915	0.7549
2	0.7712	0.7162
3	0.7931	0.7585

4.5 Comparison Details

After we implement our model, we put some reviews into model, and compare two model different behavior. Details are as follow.

Table 6 Sentiment Examples

	Great food but the service was dreadful!	**The staff should be a bit more friendly.**
Aspect	Service	Staff
AEN-Prediction	Positive	Positive
Our-Model	Negative	Negative
Label	Negative	Negative

For the first one, the wrong prediction is because machine focus on the great, which decorates food, but it ignores the adversative word 'but'.
For the second one, the AEN network failed to capture the information of more.

Whereas our model could predict both sentiment of aspect to be the right one. Again, it illustrates our model is better than AEN-BERT.

5 Conclusion

In this work, we propose an attention encoded LSTM network for the aspect sentiment analysis. We also use RoBERTa as the pretrained model. Finally, Experiment and analysis demonstrate the effectiveness and lightweight of the model.

References

[1] Fan, F., Feng, Y., & Zhao, D. (2018). Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3433-3442).

[2] Chen, P., Sun, Z., Bing, L., & Yang, W. (2017, September). Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 conference on empirical methods in natural language processing (pp. 452-461).

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[4] Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019). Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[6] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27-35.

[7] Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K. (2014, June). Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 49-54).

[8] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2019). Transformers: State-of-the-art Natural Language Processing. arXiv preprint arXiv:1910.03771.