Transformer and Multi-scale Convolution for Target-Oriented Sentiment Analysis

Yinxu Pan[1,2,3](✉), Binheng Song[1,2], Ningqi Luo[1,2], Xiaojun Chen[3], and Hengbin Cui[3]

[1] Department of Computer Science and Technology,
Tsinghua University, Beijing, China
{pyx16,lnq16}@mails.tsinghua.edu.cn
[2] Graduate School at ShenZhen, Tsinghua University, Beijing, China
songbinheng@sz.tsinghua.edu.cn
[3] Ant Financial Services Group, Xihu, China
{tiandan.cxj,alexcui.chb}@antfin.com

Abstract. Target-oriented sentiment analysis aims to extract the sentiment polarity of a specific target in a sentence. In this paper, we propose a model based on transformers and multi-scale convolutions. The transformer which is based solely on attention mechanisms generalizes well in many natural language processing tasks. Convolution layers with multiple filters can efficiently extract n-gram features at many granularities on each receptive field. We conduct extensive experiments on three datasets: SemEval ABSA challenge Restaurant and Laptop dataset, Twitter dataset. Our framework achieves state-of-the-art results, including improving the accuracy of Restaurant dataset to 84.20% (5.81% absolute improvement), improving the accuracy of the Laptop dataset to 78.21% (4.23% absolute improvement), and improving the accuracy of the Twitter dataset to 72.98% (0.87% absolute improvement).

Keywords: Target-oriented sentiment analysis · Multi-scale convolution · Transformer

1 Introduction

Target-oriented sentiment analysis is also known as aspect-term sentiment analysis, which is a central concern of the semantic web and the computational linguistics in recent years [3,8]. The goal of target-oriented sentiment analysis is to identify the sentiment polarity (i.e., negative, neutral, or positive) of a specific opinion target expressed in a comment or review by a reviewer. For example, in the sentence "*great food but the service was dreadful!*", the polarity of target "*food*" is positive while the polarity of target "*service*" is negative.

Recent years have witnessed the significant advances in sentiment analysis. For the task of target-oriented sentiment analysis, most of the previous models focus on three methods: attention mechanism, gate mechanism, and memory network.

J. Shao et al. (Eds.): APWeb-WAIM 2019, LNCS 11642, pp. 310–321, 2019.
https://doi.org/10.1007/978-3-030-26075-0_24

RNN with attention mechanism is becoming the most popular technique for this task. For example, ATAE-LSTM [16] is an LSTM model taking both the target embedding and sentence embedding as input and using an attention mechanism to calculate the representation of the sentence; IAN [7] calculates the target representation and the sentence representation in an interactive way using attention.

There are two categories of gate mechanism: the first separates the sentence using the target words, and uses gates to control the importance of each part, the second uses gates to control the importance of each word. In the first category, BILSTM-ATT-G [6] splits a sentence into three sections including the target, its left contexts and its right contexts, and then represents words in the input using a bidirectional gated recurrent neural network, and then uses three-way gated neural network structure to model the interaction between the target and its left and right contexts. In the second category, ACGE [17] is a model based on convolution neural networks and gating mechanism, which is more accurate and efficient. The gating mechanism can selectively output the sentiment features according to the given target.

Memory networks make use of multiple computational layers to capture the importance of each word. MemNet [12] takes the embedding of sentence words as memories, each layer of the MemNet is a typical attention layer. To overcome the shortcoming of attention mechanism that it cannot capture position information, MemNet employs a position weight. Based on MemNet, RAM [14] uses BiLSTM output as memories. A GRU cell is used to process the representation of each layer in RAM. TNet [5] can also be viewed as a variant of MemNet. It proposes a component to generate a target-specific representation of words in the sentence as well as incorporates a mechanism for preserving the original contextual information from the RNN layer.

These models use LSTM or CNN to process the sentence. LSTM is a sequential model, which is hard to be parallelized. And a single CNN layer does not connect all pairs of input and output positions. To tackle this problem, we propose a model based on transformer and multi-scale convolution. Transformer [15] is solely based on attention mechanisms. Previous works show that language model which is pre-trained on large corpus can generalize well in many different natural language processing tasks. Based on the transformers, OpenAI GPT [10] is one of the best pre-trained language models. We use OpenAI GPT [10] to get the representation of each word. Instead of using the representation of the start token for classification and ignoring the representations of the other tokens, we use a multi-scale convolution to extract n-gram features at many granularities. We conduct extensive experiments on SemEval 2014 Restaurants and Laptops dataset [3] and Twitter dataset, and our framework achieves state-of-the-art results, including improving the accuracy of restaurant dataset to 84.20% (5.81% absolute improvement), improving the accuracy of laptop dataset to 78.21% (4.23% absolute improvement), and improving the accuracy of twitter dataset to 72.98% (0.87% absolute improvement).

Here are our main contributions:

First, we use transformer encoders instead of LSTM or CNN encoders to improve the generalization ability of the model.
Second, multi-scale convolution can learn variable n-gram features flexibly.
Third, our model gets state-of-the-art performance on widely used datasets.

2 Transformer and Multi-scale Convolution for Target-Oriented Sentiment Analysis

In this section, we will introduce our model which combines transformer and multi-scale convolution. Figure 1 shows the structure of our model. We concatenate the sentence and the target. A multi-layer transformer is used the process the input. The multi-scale convolution learns variable n-gram features flexibly. We begin this section by presenting the problem formulation, and then describe the structure of the transformer. At last, we talk about the multi-scale convolution structure.

2.1 Problem Formulation

The input of this task is a target-sentence pair (w^{τ}, w), where target $w^{\tau} = \{w_1^{\tau}, w_2^{\tau}, ...w_m^{\tau}\}$ is a subsequence of sentence $w = \{w_1, w_2, ...w_n\}$. The aim of target-oriented sentiment analysis is to predict the sentiment polarity y of the sentence w over the target w^{τ}, where $y \in \{P, N, O\}$, P, N, O denotes "*positive*", "*negative*" and "*neutral*", respectively. For example, the sentiment polarity of sentence "*great food but the service was dreadful!*" towards target "*food*" is positive, while the sentiment polarity towards target "*service*" is negative.

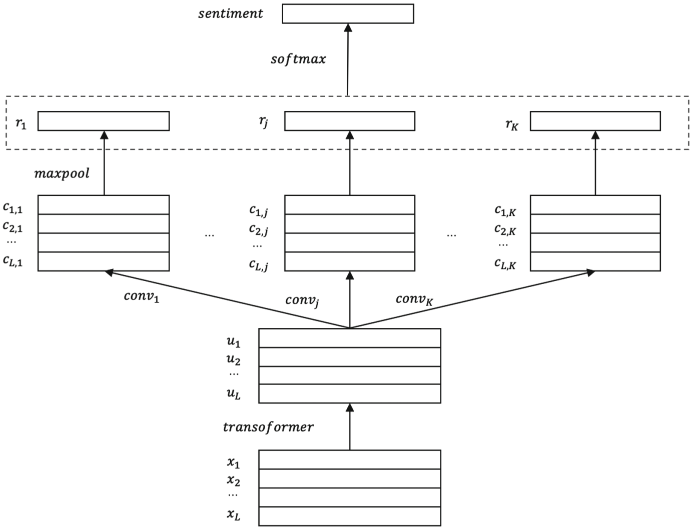

Fig. 1. Model structure

2.2 Transformer

Recurrent and convolution layers are commonly used for mapping one variable-length sequence of symbol representations to another sequence of equal length. Recurrent layers typically factor computation along the symbol positions of the input and the output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. A single convolution layer does not connect all pairs of input and output positions, which makes the path length between long-range dependencies in convolution layer much longer.

Transformer [15] is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence aligned RNN or convolution. In this part, we will give a brief introduction to the transformer.

Transformer follows an overall architecture using stacked self-attention and point-wise, fully connected layers, shown in Fig. 2.

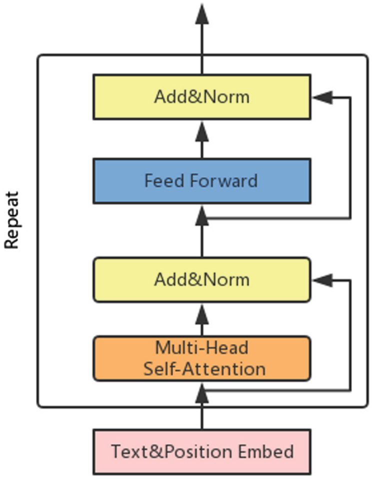

Fig. 2. Structure of transformer

Dot-Product Attention. The scaled dot-product attention is different from the conventional attention mechanisms as its attention weights are computed by a dot-product operation. Given queries $Q \in R^{T_q \times d_k}$, keys $K \in R^{T_v \times d_k}$ and values $V \in R^{T_v \times d_v}$, the attention output is

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where T_q is the sequence length of queries and T_v is the sequence length of queries and T_v is the sequence length of keys and values. d_k is the vector dimension of queries and keys, and d_v is the vector dimension of values. The sequence length of keys T_v and the sequence length of values T_v are equal. The vector dimension of queries d_k and the vector dimension of keys d_k are equal. $\sqrt{d_k}$ is used for scaling here, which guarantees the numerical stability.

Multi-head Attention. Instead of performing a single attention function with keys, values, and queries, we linearly project the queries, keys, and values h times with different, learned linear projections. 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.

$$\begin{gathered} MultiHead(Q,K,V) = Concat(head_1, ..., head_h)W^O \\ where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{gathered} \tag{2}$$

where the projections are fully connected layers without biases. $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$ and $W_O \in R^{hd_v \times d_{model}}$ are the parameters to be learned, where d_{model} is the dimension of token embeddings.

Position-Wise Feed-Forward Networks. In addition to attention sub-layers, each of the layers in a transformer block contains a fully connected feed-forward network, which consists of two linear transformations with a ReLU activation in between.

$$FFN(x) = relu(xW_1 + b_1)W_2 + b_2 \tag{3}$$

Embeddings and Softmax. Similarly to other sequence transduction models, we use learned embeddings to convert the input tokens and output tokens to vectors of dimension d_{model}. We also use a learned linear transformation and softmax function to convert the decoder output to predicted next-token probabilities. In the embedding layers, we multiply those weights by $\sqrt{d_{model}}$.

Positional Encoding. Since the multi-head attention and feed-forward network contain no convolution layers or recurrent cells, positional encoding is essential for leveraging the relative position information in sequence. We use learned position embedding instead of the sinusoidal version proposed in the original work [10].

In this paper, we use a language model which is based on transformers, and the language model is pre-trained on large corpus without supervision. The output of the language model is used as auxiliary features, and then we fine-tune it on target-oriented sentiment analysis task with supervision [10].

2.3 Multi-scale Convolution

We regard the target-oriented sentiment analysis task as a sequence pair classification problem. We concatenate the sentence $w = (w_1, w_2, ..., w_n)$ and the target $w^\tau = (w_1^\tau, w_2^\tau, ..., w_m^\tau)$, and then add a start token *start* at the beginning, add an end token *end* at the end, add a delimiter token *del* between the sentence and the target. The concatenated sequence is denoted as X, where $X = (start, w_1, w_2, ..., w_n, del, w_1^\tau, w_2^\tau, ..., w_m^\tau, end) = (x_1, x_2, ..., x_L)$. X is fed into a multi-layer transformer to get the representations of each token, where $L = m + n + 3$. For simplicity, the multi-layer transformer can be denoted as follows.

$$u_1, u_2, ..., u_L = transformer(x_1, x_2, ..., x_L) \tag{4}$$

Typically, the representation of the start token u_1 is used to represent the sentence and the target. This method ignores the representations of the other tokens, which may be helpful to predict the correct sentiment polarity. To tackle this problem, we use a multi-scale convolution to make use of the representations of all the tokens. Instead of using a single convolution layer, we use a multi-scale convolution. In the sentence, there are phrases of different lengths. To extract the representation of phrases in different length, we use convolution layers with different filter sizes. We pad the input sequence to keep the length of the output sequence same as the input sequence. K is the number of different filter sizes.

$$c_{i,j} = tanh(u_{i:i+k_j} * W_j + b_j) \tag{5}$$

where W_j and b_j are the parameters of the j-th convolution, k_j is the filter size of the j-th convolution. A max pooling layer is then used to get the most important feature.

$$r_j = max(c_{1,j}, ..., c_{i,j}, ..., c_{L,j}) \tag{6}$$

We concatenate all the max pooling outputs and get the final representation of the input.

$$r = concat(r_1, ..., r_j, ..., r_K) \tag{7}$$

At last, a fully connected layer with softmax activation is used to predict the sentiment polarity of this sentence.

$$\hat{y} = softmax(Wr + b) \tag{8}$$

where W and b are the parameters of the fully connected layer.

2.4 Loss Function

The model is trained by minimizing the sum of the language model loss and classification loss.

$$Loss = Loss_{lm} + Loss_{clf} \tag{9}$$

$\hat{x_i}$ is the i-th output of the language model, where $i \in [1, L]$. The language model loss is

$$Loss_{lm} = \sum_i \sum_n \hat{x}_{i,n} log(x_{i,n}) \tag{10}$$

where n is the index of a data sample, i is the index of word. The classification loss is a cross-entropy loss between the ground-truth y and the predicted value $\hat{y}$ for all data samples.

$$Loss_{clf} = \sum_{i} \sum_{j} \hat{y}_{i,j} log(y_{i,j}) \tag{11}$$

where i is the index of a data sample, j is the index of a sentiment class.

3 Experiments

We describe the experimental setting and report experimental results in this section.

3.1 Experimental Setting

We conduct experiments on three datasets: Restaurant and Laptop are from SemEval ABSA challenge [3]; Twitter is a target-oriented sentiment analysis dataset [2] which contains twitter posts. Table 1 shows the statistics of these datasets. These datasets are the most widely used datasets in target-oriented sentiment analysis. Following other previous work [5,11], we remove some examples having the "conflict" label. All tokens are lowercased, and we do not remove any stop words, symbols or digits. All the sentences are padded to the max length using token "PAD". Accuracy and macro-averaged F1 score are used as the main evaluation metric. For each class, precision is defined as $P = \frac{TP}{TP+FN}$, recall is defined as $R = \frac{TP}{TP+FP}$ and the F1 score is computed by $\frac{2PR}{P+R}$. TP, TN, FN and FP are the number of true positives, true negatives, false negatives, and false positives, respectively. The macro-averaged F1 score is the average F1 score across all classes [13].

Table 1. Statistics of the datasets.

Dataset	Positive	Negative	Neutral
Laptop-Train	994	870	464
Laptop-Test	341	128	169
Restaurant-Train	2164	807	637
Restaurant-Test	728	196	196
Twitter-Train	1567	1563	3127
Twitter-Test	174	174	346

Our model is compared with the following models:

- Majority is a fundamental method which assigns the majority sentiment label in training set to each instance in the test set.

- SVM uses a traditional support vector machine using n-gram features, parse features and lexicon features [4].
- AE-LSTM is a simple LSTM model that takes the target embedding and sentence embedding as input [16].
- ATAE-LSTM extends AE-LSTM with attention mechanism [16].
- IAN calculates the target representation and sentence representation in an interactively way [7].
- BILSTM-ATT-G uses gates to measure the importance of the left context and the right context, which is separated by the target [6].
- ACSA-GCAE is a model based on convolution neural networks and gating mechanisms, which is more accurate and efficient [17].
- MemNet takes the word embeddings as memory and uses a multi-layer attention mechanism to get the final representation of the sentence. Attention mechanism cannot capture position information, MemNet also uses position weights to overcome this shortcoming [12].
- RAM is an extension of the MemNet model. Different from MemNet, it uses BiLSTM hidden states as memory. A GRU cell is used to process the sentence representation of each layer. A different position weight is used for RAM [14].
- TNet propose a component to generate target-specific representations of words in the sentence, meanwhile incorporate a mechanism for preserving the original contextual information from the RNN layer [5].

We re-implemented most of these baseline models using pytorch[1] to make their results as similar as possible to those in the original papers. Each model is optimized independently. For these models, we use pre-trained GloVe vectors [9] to initialize the word embeddings and the embedding dimension is 300. We follow the parameter settings in the original paper of the baselines. The implementation of these models are available[2].

We use a 12-layer transformer with self-attention heads (768 dimensional states and 12 attention heads). We first load the pre-trained weights from OpenAI GPT, and then we fine-tune the parameters with the follow-up structures. We use five different filter sizes (from 1 to 5) for the multi-scale convolution layer. And the convolution channel is set to 100. We use Adam optimizer, and the learning rate is set to 6.25e-5. The model gets the best result within 20 epochs. The source code of our model is open and available[3].

3.2 Main Result

Table 2 shows the main results of our experiments. As the table shows, our model gets the best performance on Restaurant, Laptop, and Twitter dataset. We get 84.20% accuracy (5.81% absolute improvement) on restaurant dataset, 78.21% accuracy (4.23% absolute improvement) on laptop dataset, and 72.98% accuracy (0.87% absolute improvement) on twitter dataset.

[1] https://pytorch.org/.

[2] https://github.com/Cppowboy/ABSC_APWEBWAIM.git.

[3] https://github.com/Cppowboy/APWEB-WAIM.git.

LSTM has the worst performance of all neural networks. ATAE-LSTM improves its performance by taking the target into account and using the attention mechanism. IAN works better because it uses two attention layers. For the twitter dataset, BILSTM-ATT-G and RAM cannot perform as efficiently as they do in restaurant and laptop dataset, because they are heavily rooted in LSTM, which is not good at processing ungrammatical sentences. TNet is a model based on LSTM and CNN, which makes it works well on all the three datasets. Different from previous models, our model is based on transformers, which can solve long term dependencies and can be easily parallelized. The multi-scale convolution layer in our model can extract multi-grained features. The transformer and multi-scale convolution structure help our model get the best performance on all the three datasets.

Table 2. Experiment results (%). The result with symbol "*" is retrieved from the original paper.

Models	Restaurant		Laptop		Twitter	
	ACC	Macro-F1	ACC	Macro-F1	ACC	Macro-F1
Majority	65.00	-	53.45	-	50.00	22.22
SVM	80.89	-	72.10	-	63.40	63.30
LSTM	76.70	63.57	69.28	63.30	66.04	63.46
ATAE-LSTM	77.23	63.73	69.44	63.46	71.24	69.19
IAN	78.60*	-	72.10*	-	-	-
BILSTM-ATT-G	79.20	67.07	71.32	64.88	71.68	70.37
GCAE	78.12	62.50	70.38	64.02	72.40	70.89
MemNet	77.86	64.47	68.18	62.46	69.80	66.86
RAM	78.30	65.42	71.63	66.73	71.24	68.75
TNet	78.39	65.37	73.98	68.64	72.11	70.01
Ours	84.20	76.35	78.21	73.31	72.98	71.40

3.3 Effectiveness of Pretraining

To show the effectiveness of the pre-training procedure, we train our model twice: the first time we use pre-trained parameters from the OpenAI GPT and then fine-tune the parameters, the second time we train the model from scratch. The experiments are conducted on the restaurant and laptop dataset. Table 3 shows the result of the experiments.

As the table shows, training from scratch works poorly, and the pre-trained work has a significant performance gain. We observe that the lack of pre-training hurts performances, resulting in a considerable performance decrease compared to our full model.

Table 3. Effectiveness of pre-training

Models	Restaurant		Laptop	
	ACC	Marco-F1	ACC	Marco-F1
w/o pre-training	69.20	48.16	64.89	59.25
w/ pre-training	84.20	76.35	78.21	73.31

3.4 Effectiveness of Multi-scale Convolution

In this part, we design a simple model without multi-scale convolution. The simplified model uses the representation of the first token as the representation of the input and uses a fully connected layer for classification. Another model we used is our full model. The experiments are conducted on the restaurant and laptop dataset. Table 4 shows the result of the experiments. As the table shows, the multi-scale convolution layer improves the performance of our model (including 0.81% improvement for the restaurant dataset and 0.78% improvement for the laptop dataset).

Table 4. Effectiveness of multi-scale CNN

Models	Restaurant		Laptop	
	ACC	Marco-F1	ACC	Marco-F1
w/o cnn	83.39	74.40	77.43	72.42
w/ cnn	84.20	76.35	78.21	73.31

3.5 Case Study

Table 5 shows some sample cases. The input targets are wrapped in the brackets with the correct labels given as subscripts. The notations P, N and O in the table represent positive, negative and neutral respectively. For example, for the target "coffee" in the first sentence, the sentiment polarity is positive.

Our model can predict target sentiment more accurately than ATAE-LSTM and GCAE. ATAE-LSTM is an LSTM based model, which relies on sequential information. It can perform well for formal sentences. For the first two sentences, ATAE-LSTM gets the correct prediction, while GCAE fails to get the correct prediction. GCAE is a CNN based model, which is good at processing ungrammatical text. For the third and the fourth sentence, GCAE get the correct prediction, while ATAE-LSTM fails to get the correct prediction. Our model is a transformer based model, which can process both grammatical and ungrammatical sentences. And our model can solve some difficult cases. For the fifth and the sixth sentence, our model can extract the sentimental relation between context words such as negation and comparison. For the last sentence, because of the multi-scale convolution layer, our model can deal with the noncompositional sentiment expression "what I go for" and make the correct prediction.

Table 5. Example predictions. The input targets are wrapped in brackets with the true labels given as subscripts. ✗ indicates incorrect prediction.

Sentence	ATAE-LSTM	GCAE	Ours
[Coffee]$_P$ is a better deal than overpriced sandwiches	P	O✗	P
But make sure you have enough room on your credit card as the [bill]$_P$ will leave a big dent in your wallet	P	O✗	P
Aww, it's okay... You have a [PSP]$_P$. :D That's good already	O✗	P	P
I hate my [iPod]$_N$! It's dead! dead dead dead! ! ! Someone wanna fix it for me?	O✗	N	N
I have never had a bad [meal]$_P$ (or bad service) at pigalle	N✗	N✗	P
The [staff]$_N$ should be a bit more friendly	P✗	P✗	N
It's a basic pizza joint, not much to look at, but the [pizza]$_P$ is what I go for	N✗	O✗	P

4 Conclusion

In this paper, we proposed a model, which combines transformer and multi-scale convolution. The transformer can solve long term dependencies and can be easily parallelized. The multi-scale convolution can extract multi-grained features. The performance of our model consistently dominates previous state-of-the-art methods on different types of data. The experiments show the efficacy of different modules of our model.

References

1. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
2. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: 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), vol. 2, pp. 49–54 (2014)
3. Kirange, D., Deshmukh, R.R.: Emotion classification of restaurant and laptop review dataset: SemEval 2014 task 4. Int. J. Comput. Appl. **113**(6) (2015)
4. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442 (2014)
5. Li, X., Bing, L., Lam, W., Shi, B.: Transformation networks for target-oriented sentiment classification. arXiv preprint arXiv:1805.01086 (2018)

6. Liu, J., Zhang, Y.: Attention modeling for targeted sentiment. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, vol. 2, pp. 572–577 (2017)
7. Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. arXiv preprint arXiv:1709.00893 (2017)
8. Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., Stoyanov, V.: SemEval-2016 task 4: sentiment analysis in twitter. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 1–18 (2016)
9. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
10. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018). https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
11. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. arXiv preprint arXiv:1512.01100 (2015)
12. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. arXiv preprint arXiv:1605.08900 (2016)
13. Tay, Y., Tuan, L.A., Hui, S.C.: Dyadic memory networks for aspect-based sentiment analysis. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 107–116. ACM (2017)
14. Tran, K., Bisazza, A., Monz, C.: Recurrent memory networks for language modeling. arXiv preprint arXiv:1601.01272 (2016)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Wang, Y., Huang, M., Zhao, L., et al.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 606–615 (2016)
17. Zhang, M., Zhang, Y., Vo, D.T.: Gated neural networks for targeted sentiment analysis. In: AAAI, pp. 3087–3093 (2016)