Extracting Opinion Expression with Neural Attention

Jiachen Du[1], Lin Gui[1], and Ruifeng Xu[1,2(✉)]

[1] Shenzhen Engineering Laboratory of Performance Robots at Digital Stage, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
dujiachenl99l65@gmail.com, guilin.nlp@gmail.com, xuruifeng.hits@gmail.com
[2] Guangdong Provincial Engineering Technology Research Center for Data Science, Guangzhou, China

Abstract. Extracting opinion expressions from raw text is a fundamental task in sentiment analysis and it is usually formulated as a sequence labeling problem tackled by conditional random fields (CRFs). However CRF-based models usually need abundant hand-crafted features and require a lot of engineering effort. Recently deep neural networks are proposed to alleviate this problem. In order to extend neural-network-based models with ability to emphasize related parts in text, we propose a novel model which introduces the attention mechanism to Recurrent Neural Networks (RNNs) for opinion expression sequence labeling. We evaluate our model on MPQA 1.2 dataset, and experimental results show that the proposed model outperforms state-of-the-art CRF-based model on this task. Visualization of some examples show that our model can make use of correlation of words in the sentences and emphasize the crucial parts for this task to improve the performance compared with the vanilla RNNs.

Keywords: Opinion expression extraction · Sequence labeling · Recurrent neural network · Neural attention

1 Introduction

Recently, researchers from many subareas of Natural Language Processing and Machine Learning have been working on the sentiment analysis and related tasks [8, 13, 14, 17, 20]. In this work, we focus on one fundamental task in sentiment analysis—the detection of opinion expressions—both direct subjective expressions (DSEs) and expressive subjective expressions (ESEs) as defined in Wiebe et al. [18]. DSEs are explicit mentions of private states or speech events expressing private states; and ESEs are expressions that indicate sentiment, emotion, etc. without explicitly conveying them.

Opinion expressions extraction has often been treated as a sequence labeling task in previous works. This approach usually uses the conventional B-I-O tagging scheme to convert the original opinion expressions to sequences of tagging tokens: B indicates the beginning of an opinion expression, I is for the token within the range of opinion expression, O is the tag used to denote token outside any opinion expression. Since two

Y. Li et al. (Eds.): SMP 2016, CCIS 669, pp. 151–161, 2016.
DOI: 10.1007/978-981-10-2993-6_13

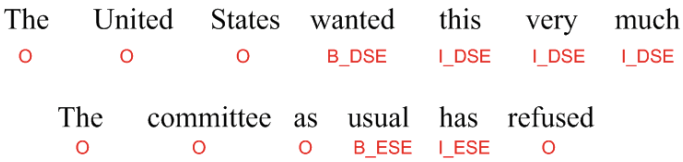

Fig. 1. Example Sentences with opinion expression B-I-O labels

types of opinion expressions (DSE, ESE) are used in annotation, there are five tagging labels in this task: B_DSE, I_DSE, B_ESE, I_ESE and O. The example sentences in Fig. 1. show this tagging scheme. For instance, the DSE "wanted this very much" results in one B_DSE tag for "wanted" and three I_DSE tags for "this very much".

Conditional random fields (CRFs) [10] have been quite successful for different sequence labeling problem in sentiment analysis including opinion target extraction [15], opinion holder recognition [11] etc. The state-of-the-art models of opinion expression extraction are also CRF [2] and variant of CRF that relaxes the Markovian assumption [21]. However, the success of CRFs depends heavily on the use of an appropriate features set and carefully manual selection, which requires a lot of engineering effort.

In recent years, there is no doubt that deep learning has ushered in amazing technological advances on natural language processing (NLP) researches. Deep learning models automatically learn the latent features and represent them as distributed vectors, outperforming CRF-based model in several tasks of NLP. For example, Yao et al. applied Recurrent Neural Network (RNN) to name entity recognition task, and showed that RNN obtains state-of-the-art result in this task [22]. Based on the aforementioned architectures, a new direction of neural networks has emerged. It learns to focus "attention" to specific parts of text as the simulation of human's attention while reading. The researches on neural network with attention mechanism show promising results on a sequence-to-sequence (seq2seq) tasks in NLP, including machine translation [1], caption generation [19] and text summarization [16].

Motivated by the recent researches on attention model of neural networks, we explore to apply recurrent neural network with attention to opinion expression extraction which can be treated as an instance of seq2seq learning tasks. In general, we except that the neural attention model would make use of correlation of words in the sentences and emphasize the crucial parts for this task to improve the performance compared with the vanilla RNNs.

The rest of this paper proceeds as follows. In Sect. 2, we present our recurrent neural network with attention model. In Sect. 3, we show the experimental results on MPQA dataset and analyze them. In Sect. 4, we conclude and discuss future work.

2 Methodology

This section describes a novel architecture for opinion expression extraction. The new architecture consists of a bidirectional recurrent neural network with long short-term memory (LSTM) as an word encoder, a decoder that outputs the predicted B-I-O tags of

opinion expressions, and a neural attention layer that softly aligns the word sequences and output sequences.

2.1 RNN with Long Short-Term Memory

An RNN [4] is a kind of neural network that processes sequences of arbitrary length by recursively applying a function to its hidden state vector $h_t \in \mathbb{R}^d$ of each element in the input sequences. The hidden state vector at time-step t depends on the input symbol x_t and the hidden state vector at last time-step h_{t-1} is:

$$h_t = \begin{cases} 0 & t = 0 \\ g(h_{t-1}, x_t) & \text{otherwise} \end{cases} \tag{1}$$

A fundamental problem in traditional RNN is that gradients propagated over many steps tend to either vanish or explode. It makes RNN difficult to learn long-dependency correlations in a sequence. Long short-term memory network (LSTM) was proposed by [7] to alleviate this problem. LSTM has three gates: an input gate i_t, a forget gate f_t, an output gate o_t and a memory cell c_t. They are all vectors in $\mathbb{R}^d$. The LSTM transition equations are:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}), \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}), \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1}), \\ \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1}), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{2}$$

where x_t is the input at the current time step, σ is the sigmoid function and $\odot$ is the elementwise multiplication operation. In our model, we use the output vector o_t of each time step as the representation of input sequence.

2.2 Bidirectional RNNs

Observe that with above definition, LSTMs only have information about the past, when making a decision on input x_t. This limits LSTMs to make use of previous sequential information which is important for most NLP tasks. To capture long-distance dependencies from the future as well as from the past, Graves and et al. proposed to use bidirectional LSTMs which allow bidirectional links in the network [6]. For the Elman-type RNN in Sect. 2.1, the bidirectional variant of it is:

$$\begin{aligned}\overrightarrow{h_t} &= \overrightarrow{g}(\overrightarrow{h_{t-1}}, x_t)(\overrightarrow{h_0} = 0)\\ \overleftarrow{h_t} &= \overleftarrow{g}(\overleftarrow{h_{t+1}}, x_t)(\overleftarrow{h_T} = 0)\\ h_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}]\end{aligned} \tag{3}$$

where $\overrightarrow{g}$ and $\overleftarrow{g}$ are forward and backward transitional functions, they use different weight matrices and bias vectors. The concatenated vector $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ combines vectors of the same time-step from both directions. We can thus interpret h_t as an intermediate representation summarizing the past and the future, which is then used to make decision on the current input. Similarly, unidirectional LSTMs can be extended to bidirectional LSTMs by allowing bidirectional connections in the hidden layers.

2.3 Stacked RNNs

Here, we describe briefly the underlying framework, called *Stacked RNNs* proposed by (El Hihi and Bengio) [3] on which we build a novel architecture that model attention. In the Stacked RNNs framework, there are $k(k > = 2)$ RNNs RNN_1, RNN_2, …, RNN_k where the jth RNN receive $(j-1)$th RNN's output as its input and feed its output into the $(j+1)$th RNN, meanwhile the first RNN receives the word sequences as its input and the last RNN omits the vector representation of the labels which are used to predict the targets. Suppose the output of j^{th} RNN on time-step t is h_t^j, the stacked RNNs can be formulated as:

$$h_t^j = \begin{cases} x_t & j = 0 \\ g(h_{t-1}^j, h_t^{j-1}) & \text{otherwise} \end{cases} \tag{4}$$

The function g used in (4) can be replaced by any RNN transition function, In this paper, we use bidirectional LSTM described in Sect. 2.2. Figure 2 demonstrates a stacked RNN consisting two LSTMs, the input sequence is the vectors of words in sentences and the output sequence is the B-I-O tags of opinion expressions. In order to make the stacked RNNs to be extended easily, we use stacked bidirectional LSTMs with depth of 2 as our basic model in this paper.

2.4 Stacked RNNs with Neural Attention

Recently, researches on neural network with attention mechanism show promising results on a sequence-to-sequence (seq2seq) tasks in NLP, including machine translation [1], caption generation [19] and text summarization [16]. For opinion expression extraction, we proposes to use neural attention to focus the important parts in the sentences. As we described in Sect. 2.3, we use stacked bidirectional-LSTMs with depth of 2 as our basic model. For the attention model, the input of the second LSTM on each time step t is a weighed sum of the first LSTM's output vectors. The input vector of the second LSTM on time t, i_t^2 is represented by

Fig. 2. Demonstration of stacked RNNs for emotion expression extraction, the input of the whole model is the word embeddings and the output is the predicted B-I-O tags. In this paper we use stacked bidirectional LSTMs with depth of 2 as our basic model

$$i_t^2 = \sum_{s=1}^{T} \alpha_{ts} h_s^1 \tag{5}$$

In Eq. (5), h_s^1 is the output vector of the 1^{st} LSTM on time step s, α_{ts} is the weight value that maps output sequence of the 1^{st} LSTM $[h_1^1, h_2^1, \ldots, h_T^1]$ to input vector of the 2^{nd} LSTM. α_{ts} can also be consider as a value that indicates how much of a difference the s^{th} word will make to the decision of the t^{th} label. The weight α_{ts} is obtained by

$$\begin{aligned} e_{ts} &= \tanh(W^1 h_s^1 + W^2 h_{t-1}^2 + b) \\ \alpha_{ts} &= \frac{\exp(e_{ts}^T e)}{\sum_{k=1}^{T} \exp(e_{tk}^T e)} \end{aligned} \tag{6}$$

In Eq. (6), W^1 and W^2 are parametric matrices that will be tuned in training phase, b is the bias vector. e in this equation is a vector with the same length with e_{ts}, and is jointly trained with all other parameters. The first line in this equation can be treated as a fully-connected neural network whose input is the output vectors of the emitted vectors of both LSTMs with separated parametric matrix. The second line in Eq. (6) is also a fully-connected neural network but with a softmax activation function that outputs the attention weights. The whole model is illustrated in Fig. 3.

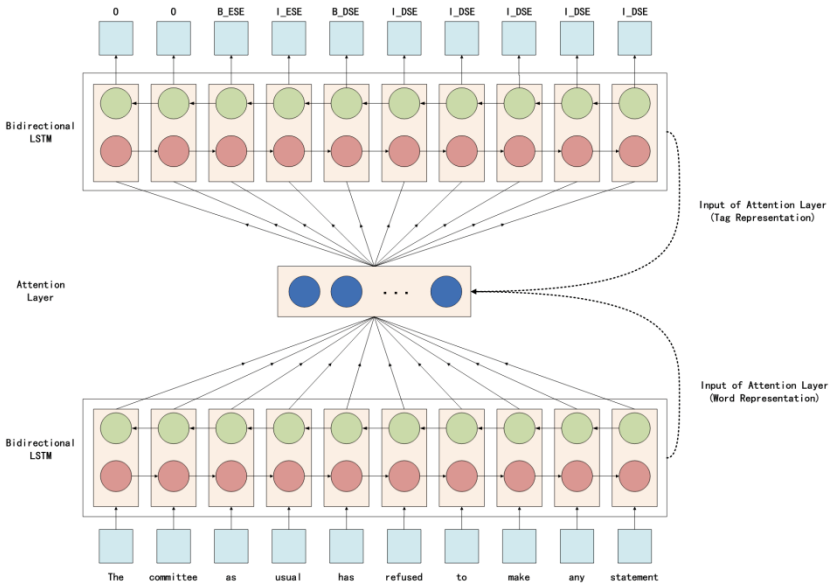

Fig. 3. Stacked RNNs with neural attention. For the sake of simplicity, the attention layer in this figure is represented by a abstract part.

3 Experiments

In this section, we investigate the empirical performance of our proposed model on opinion expression extraction and compare it with state-of-the-art models for this task. We use MPQA 1.2 corpus[1] [18]. It contains 535 news documents of 11,111 sentences annotated with both DSEs and ESEs labels at phrase level. As in previous work, we use 135 documents as a development set and employ 10-fold cross validation on the remaining 400 documents. The summary statistics of MPQA 1.2 is listed in Table 1.

Table 1. Summary statistics of the MPQA 1.2 datasets.

	DSE	ESE
Sentences with opinion (%)	55.89	57.93
Words with opinion (%)	5.82 %	8.44 %
Maximum of Length	15	40
Minimum of Length	1	2
Average of Length	1.86	3.33

[1] Available at http://www.cs.pitt.edu/mpqa/.

3.1 Evaluation Metrics

We use precision, recall, and F1-measure to evaluate the performance of the model. Since the boundaries of opinion expressions are hard to define even for human annotators [18], we use *Binary Overlap* and *Proportional Overlap* as two soft measures to evaluate the performance. Breck et al. firstly introduced the *Binary Overlap* measure to opinion expression extraction which counts every overlapping match between a predicted and true expression as correct [2]. And *Proportional Overlap* is a stricter measure that computes the proportion of overlapping spans [9].

3.2 Model Training and Hyper-parameters

The model can be trained in an end-to-end way by back-propagation, where the objective function is cross-entropy of error loss. Training is done through gradient descent with the Adadelta update rule. In all of these experiments, the word embeddings are initialized with the publicly available word2vec vectors that were trained on 100 billion words from Google News [12]. Other parameters are set as follows. The number of hidden units of both LSTM is 32, dropout rate is 0.5 and mini-batch size is 128. These hyper-parameters are chosen via a grid search on the development set.

3.3 Baselines

To illustrate the performance boost of our proposed attention model, we compare our model with some baseline methods. Since we use bidirectional LSTM as component of our model, we implement an RNN with LSTM memory unit as a baseline. We also compare our model with stacked LSTM with depth of 2.

- **Bi-LSTM:** LSTM for sequence labelling. [5]
- **Bi-LSTM(stacked):** stacked model of two bi-directional LSTMs [8].

We also compare our model with the following state-of-the-art models:

- **CRF:** Features used in CRF are words, part-of-speech tags and membership in a manually constructed opinion lexicon (within a [−1,+1] context window) [2].
- **Semi-CRF:** Since Semi-CRF is a variant of traditional CRF model that relaxes the Markovian assumption and focus on the phrase level features rather than token-level features. Semi-CRF also use parse trees to generate the candidate segments of sentences [21].

3.4 Results and Analysis

Since our model is based on RNNs, we firstly conduct experiments to confirm that our model outperforms vanilla bidirectional LSTM and stacked LSTM. The experimental results are shown in Table 2. We notice that vanilla bidirectional LSTM performs the worst among all the models since it cannot extract high-level features for this task. Two-layer LSTM uses deeper architecture “in space” to give LSTM additional power

to tackle complex problems, and it obtains higher F1 scores than the vanilla LSTM. Our model which introduces the attention layer to stacked LSTM gives the best performance among the three models. For F1 scores, our model outperforms stacked LSTM with maximum absolute gains of 2.80 % for DSE, and 3.39 % for ESE. All differences are statistically significant at the 0.05 level. These results can demonstrate that neural attention model can emphasize the crucial parts for specific tasks and improve the performance of RNNs on sequence labeling problems.

Table 2. Experimental evaluation of our proposed model and baseline methods

Task	Model	P		R		F1	
		Bin	Prop	Bin	Prop	Bin	Prop
DSE	Bi-LSTM	64.31	61.21	70.90	65.33	67.44	62.25
	Bi-LSTM(stacked)	64.80	63.22	72.15	65.35	68.27	63.28
	Bi-LSTM(stacked) + Att	**67.82**	**64.32**	**74.89**	**65.89**	**71.17**	**65.10**
ESE	Bi-LSTM	56.34	48.20	70.00	52.18	62.43	50.11
	Bi-LSTM(stacked)	57.10	48.37	70.48	54.20	63.09	51.12
	Bi-LSTM(stacked) + Att	**63.29**	**48.69**	**70.02**	**55.97**	**66.48**	**52.06**

Table 3 shows comparison of our model to the previous best results in the literature. In term of F1 value, our model performs best for both DSE and ESE detection. Semi-CRF with its high recall, performs comparably to our model on F1 measure. Note that our model does not have to access any hand-crafted features other than word embeddings pre-trained by word2vec. In general, CRF models achieve high precision but low recall on both DSE and ESE detection (Note that it obtains best precision for binary and proportional measures, however it performs worst for recall measure). While Semi-CRF exhibit a high recall, low precision performance, since it use a more relaxed Marcovian assumption. Compared with Semi-CRF, our model produces even higher recall and comparable precision. We can observe that our model obtains higher F1 scores than Semi-CRF — 71.17 vs. 71.15 (binary overlap) and 65.10 vs. 64.27 (proportional overlap) for DSEs; 66.48 vs. 66.37 (binary overlap) and 57.57 vs. 50.95 (proportional overlap) for ESEs.

Table 3. Results of our proposed model against CRF-based models.

Task	Model	P		R		F1	
		Bin	Prop	Bin	Prop	Bin	Prop
DSE	CRF	**82.28**	**74.96**	52.99	46.98	64.45	57.74
	Semi-CRF	69.41	61.67	73.08	**67.22**	71.15	64.27
	Our Model	67.82	64.32	**74.89**	65.89	**71.17**	**65.10**
ESE	CRF	68.36	**56.08**	51.84	42.26	58.85	48.10
	Semi-CRF	**69.06**	45.64	64.15	**58.05**	66.37	50.95
	Our Model	63.29	**48.69**	**70.02**	55.97	**66.48**	**52.06**

3.5 Case Study

In order to validate that our model is able to select salient parts in a text sequence, we visualize the attention layers in Fig. 4. For an example sentence from the MPQA dataset in which our model predicted all labels correctly. The example sentence and its corresponding labels are:

Nevertheless	*he*	*wanted*	*to*	*clarify*	*some*	*of*	*Powell*	*'s*	*statement*
O	***O***	***B_DSE***	***B_DSE***	***B_DSE***	***O***	***O***	***O***	***O***	***O***

This sentence contains a DSE "*wanted to clarify*" which is a verb phrase. In order to understand the attitudes and feelings which this phrase conveys, we have to consider its corresponding object — "*Powell's statement*". We except our attention model can recognize this correlation and emphasize it for extracting the correct opinion expressions.

In Fig. 4. deeper colors mean higher attention and pale colors indicate lower attention. First of all, we can observe that for each label, the highest attention value is always associate with its corresponding word in the sentence. This result is consistent to our expectation, since each word has the biggest influence on its corresponding label. We can also find that except *"wanted to clarify"* 's own words, the phrase "*Powell's statement*" has the most highest attention value on the labels of this DSE. This means our model can emphasize words related to the opinion expressions other than the corresponding ones in text. This example shows that introducing attention mechanism gives RNNs additional power to tackle more complicated sequence labeling problems that involve semantic understanding.

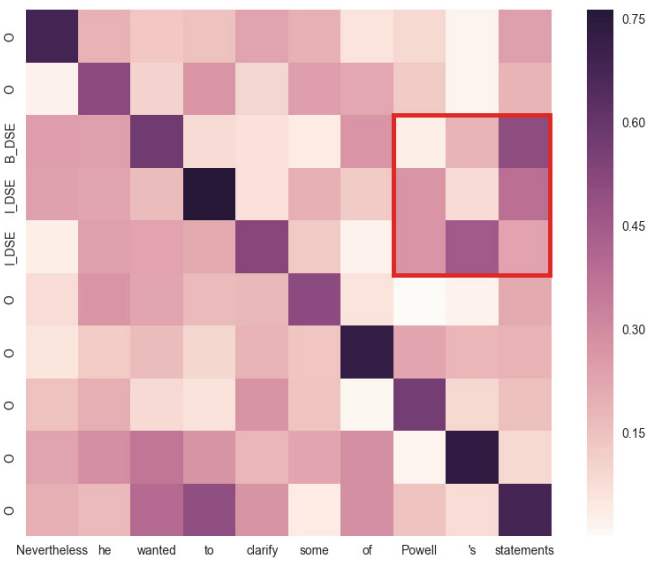

Fig. 4. Visualization of attention signals in sample sentences in the MPQA dataset. (Color figure online)

4 Conclusion

In this paper, we improve the traditional recurrent neural networks (RNNs) by introducing the attention mechanism to tackle the opinion expression extraction task. The new model can emphasize the most important parts in text and evaluate the correlation of each words in sentence with their expression labels (DSE and ESE). Experimental results show that attention layer gives RNNs additional power to process more complicated sequence labeling problems such as opinion expression extraction. Since our model can produce higher recall on both DSE and ESE, it outperforms traditional CRF-based methods on MPQA dataset.

In the future, we would like apply our models to other sequence labeling tasks in sentiment analysis including opinion holder extraction, aspect-based sentiment analysis, etc.

Acknowledgement. This work was supported by the National Natural Science Foundation of China 61370165, 61632011, National 863 Program of China 2015AA015405, Shenzhen Peacock Plan Research Grant KQCX20140521144507925 and Shenzhen Foundational Research Funding JCYJ20150625142543470, Guangdong Provincial Engineering Technology Research Center for Data Science 2016KF09.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Breck, E., Choi, Y., Cardie, C.: Identifying expressions of opinion in context. In: IJCAI, pp. 2683–2688 (2007)
3. El Hihi, S., Bengio, Y.: Hierarchical recurrent neural networks for long-term dependencies. In: NIPS, p 409. Citeseer (1995)
4. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**, 179–211 (1990)
5. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
6. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**, 602–610 (2005)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
8. Irsoy, O., Cardie, C.: Opinion mining with deep recurrent neural networks. In: EMNLP, pp. 720–728 (2014)
9. Johansson, R., Moschitti, A.: Syntactic and semantic structure for opinion expression detection. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp. 67–76. Association for Computational Linguistics (2010)
10. Lafferty, J., Mccallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML, pp. 282–289 (2001)
11. Lu, B.: Identifying opinion holders and targets with dependency parser in Chinese news texts. In: Proceedings of the NAACL HLT 2010 Student Research Workshop, pp. 46–51. Association for Computational Linguistics (2010)

12. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
13. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p 271. Association for Computational Linguistics (2004)
14. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. Association for Computational Linguistics (2002)
15. Pontiki, M., Galanis, D., Papageorgiou, H., et al.: Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, pp. 486–495. Association for Computational Linguistics (2015)
16. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
17. Tang, D., Wei, F., Yang, N., et al.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: ACL (1), pp. 1555–1565 (2014)
18. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Lang. Resour. Eval. **39**, 165–210 (2005)
19. Xu, K., Ba, J., Kiros, R., et al.: Show, attend and tell: neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044 2:5 (2015)
20. Xu, R., Gui, L., Xu, J., et al.: Cross lingual opinion holder extraction based on multi-kernel SVMs and transfer learning. World wide web **18**, 299–316 (2015)
21. Yang, B., Cardie, C.: Extracting opinion expressions with semi-Markov conditional random fields. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1335–1345. Association for Computational Linguistics (2012)
22. Yao, K., Zweig, G., Hwang, M.-Y., et al: Recurrent neural networks for language understanding. In: INTERSPEECH, pp. 2524–2528 (2013)