Attention-based LSTM for Aspect-level Sentiment Classification

Yequan Wang and **Minlie Huang** and **Li Zhao*** and **Xiaoyan Zhu**
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
*Microsoft Research Asia
wangyequan@live.cn, aihuang@tsinghua.edu.cn
lizo@microsoft.com, zxy-dcs@tsinghua.edu.cn

Abstract

Aspect-level sentiment classification is a fine-grained task in sentiment analysis. Since it provides more complete and in-depth results, aspect-level sentiment analysis has received much attention these years. In this paper, we reveal that the sentiment polarity of a sentence is not only determined by the content but is also highly related to the concerned aspect. For instance, "*The appetizers are ok, but the service is slow.*", for aspect *taste*, the polarity is positive while for *service*, the polarity is negative. Therefore, it is worthwhile to explore the connection between an aspect and the content of a sentence. To this end, we propose an Attention-based Long Short-Term Memory Network for aspect-level sentiment classification. The attention mechanism can concentrate on different parts of a sentence when different aspects are taken as input. We experiment on the SemEval 2014 dataset and results show that our model achieves state-of-the-art performance on aspect-level sentiment classification.

1 Introduction

Sentiment analysis (Nasukawa and Yi, 2003), also known as opinion mining (Liu, 2012), is a key NLP task that receives much attention these years. Aspect-level sentiment analysis is a fine-grained task that can provide complete and in-depth results. In this paper, we deal with aspect-level sentiment classification and we find that the sentiment polarity of a sentence is highly dependent on both content and aspect. For example, the sentiment polarity of "*Staffs are not that friendly, but the taste covers all.*" will be positive if the aspect is *food* but negative when considering the aspect *service*. Polarity could be opposite when different aspects are considered.

Neural networks have achieved state-of-the-art performance in a variety of NLP tasks such as machine translation (Lample et al., 2016), paraphrase identification (Yin et al., 2015), question answering (Golub and He, 2016) and text summarization (Rush et al., 2015). However, neural network models are still in infancy to deal with aspect-level sentiment classification. In some works, target dependent sentiment classification can be benefited from taking into account target information, such as in Target-Dependent LSTM (TD-LSTM) and Target-Connection LSTM (TC-LSTM) (Tang et al., 2015a). However, those models can only take into consideration the target but not aspect information which is proved to be crucial for aspect-level classification.

Attention has become an effective mechanism to obtain superior results, as demonstrated in image recognition (Mnih et al., 2014), machine translation (Bahdanau et al., 2014), reasoning about entailment (Rocktäschel et al., 2015) and sentence summarization (Rush et al., 2015). Even more, neural attention can improve the ability to read comprehension (Hermann et al., 2015). In this paper, we propose an attention mechanism to enforce the model to attend to the important part of a sentence, in response to a specific aspect. We design an aspect-to-sentence attention mechanism that can concentrate

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615,
Austin, Texas, November 1-5, 2016.

on the key part of a sentence given the aspect.

We explore the potential correlation of aspect and sentiment polarity in aspect-level sentiment classification. In order to capture important information in response to a given aspect, we design an attention-based LSTM. We evaluate our approach on a benchmark dataset (Pontiki et al., 2014), which contains restaurants and laptops data.

The main contributions of our work can be summarized as follows:

- We propose attention-based Long Short-Term memory for aspect-level sentiment classification. The models are able to attend different parts of a sentence when different aspects are concerned. Results show that the attention mechanism is effective.
- Since aspect plays a key role in this task, we propose two ways to take into account aspect information during attention: one way is to concatenate the aspect vector into the sentence hidden representations for computing attention weights, and another way is to additionally append the aspect vector into the input word vectors.
- Experimental results indicate that our approach can improve the performance compared with several baselines, and further examples demonstrate the attention mechanism works well for aspect-level sentiment classification.

The rest of our paper is structured as follows: Section 2 discusses related works, Section 3 gives a detailed description of our attention-based proposals, Section 4 presents extensive experiments to justify the effectiveness of our proposals, and Section 5 summarizes this work and the future direction.

2 Related Work

In this section, we will review related works on aspect-level sentiment classification and neural networks for sentiment classification briefly.

2.1 Sentiment Classification at Aspect-level

Aspect-level sentiment classification is typically considered as a classification problem in the literature. As we mentioned before, aspect-level sentiment classification is a fine-grained classification task. The majority of current approaches attempt to detecting the polarity of the entire sentence, regardless of the entities mentioned or aspects. Traditional approaches to solve those problems are to manually design a set of features. With the abundance of sentiment lexicons (Rao and Ravichandran, 2009; Perez-Rosas et al., 2012; Kaji and Kitsuregawa, 2007), the lexicon-based features were built for sentiment analysis (Mohammad et al., 2013). Most of these studies focus on building sentiment classifiers with features, which include bag-of-words and sentiment lexicons, using SVM (Mullen and Collier, 2004). However, the results highly depend on the quality of features. In addition, feature engineering is labor intensive.

2.2 Sentiment Classification with Neural Networks

Since a simple and effective approach to learn distributed representations was proposed (Mikolov et al., 2013), neural networks advance sentiment analysis substantially. Classical models including Recursive Neural Network (Socher et al., 2011; Dong et al., 2014; Qian et al., 2015), Recursive Neural Tensor Network (Socher et al., 2013), Recurrent Neural Network (Mikolov et al., 2010; Tang et al., 2015b), LSTM (Hochreiter and Schmidhuber, 1997) and Tree-LSTMs (Tai et al., 2015) were applied into sentiment analysis currently. By utilizing syntax structures of sentences, tree-based LSTMs have been proved to be quite effective for many NLP tasks. However, such methods may suffer from syntax parsing errors which are common in resource-lacking languages.

LSTM has achieved a great success in various NLP tasks. TD-LSTM and TC-LSTM (Tang et al., 2015a), which took target information into consideration, achieved state-of-the-art performance in target-dependent sentiment classification. TC-LSTM obtained a target vector by averaging the vectors of words that the target phrase contains. However, simply averaging the word embeddings of a target phrase is not sufficient to represent the semantics of the target phrase, resulting a suboptimal performance.

Despite the effectiveness of those methods, it is still challenging to discriminate different sentiment polarities at a fine-grained aspect level. Therefore, we are motivated to design a powerful neural network which can fully employ aspect information for sentiment classification.

3 Attention-based LSTM with Aspect Embedding

3.1 Long Short-term Memory (LSTM)

Recurrent Neural Network(RNN) is an extension of conventional feed-forward neural network. However, standard RNN has the gradient vanishing or exploding problems. In order to overcome the issues, Long Short-term Memory network (LSTM) was developed and achieved superior performance (Hochreiter and Schmidhuber, 1997). In the LSTM architecture, there are three gates and a cell memory state. Figure 1 illustrates the architecture of a standard LSTM.

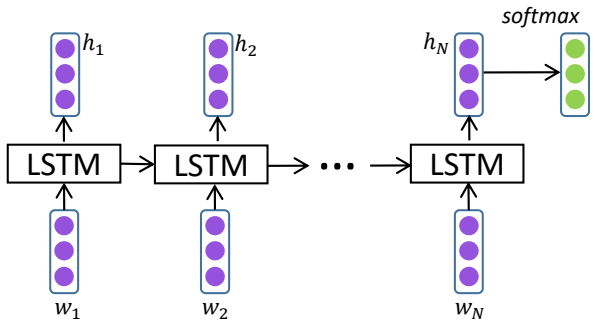


Figure 1: The architecture of a standard LSTM. $\{w_1, w_2, \ldots, w_N\}$ represent the word vector in a sentence whose length is N. $\{h_1, h_2, \ldots, h_N\}$ is the hidden vector.

More formally, each cell in LSTM can be computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \quad (1)$$

$$f_t = \sigma(W_f \cdot X + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot X + b_i) \quad (3)$$

$$o_t = \sigma(W_o \cdot X + b_o) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c) \quad (5)$$

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

where $W_i, W_f, W_o \in \mathbb{R}^{d \times 2d}$ are the weighted matrices and $b_i, b_f, b_o \in \mathbb{R}^d$ are biases of LSTM to be learned during training, parameterizing the transformations of the input, forget and output gates respectively. σ is the sigmoid function and $\odot$ stands for element-wise multiplication. x_t includes the inputs of LSTM cell unit, representing the word embedding vectors w_t in Figure 1. The vector of hidden layer is h_t.

We regard the last hidden vector h_N as the representation of sentence and put h_N into a $softmax$ layer after linearizing it into a vector whose length is equal to the number of class labels. In our work, the set of class labels is $\{$*positive*, *negative*, *neutral*$\}$.

3.2 LSTM with Aspect Embedding (AE-LSTM)

Aspect information is vital when classifying the polarity of one sentence given aspect. We may get opposite polarities if different aspects are considered. To make the best use of aspect information, we propose to learn an embedding vector for each aspect.

Vector $v_{a_i} \in \mathbb{R}^{d_a}$ is represented for the embedding of aspect i, where d_a is the dimension of aspect embedding. $A \in \mathbb{R}^{d_a \times |A|}$ is made up of all aspect embeddings. To the best of our knowledge, it is the first time to propose aspect embedding.

3.3 Attention-based LSTM (AT-LSTM)

The standard LSTM cannot detect which is the important part for aspect-level sentiment classification. In order to address this issue, we propose to design an attention mechanism that can capture the key part of sentence in response to a given aspect. Figure 2 represents the architecture of an Attention-based LSTM (AT-LSTM).

Let $H \in \mathbb{R}^{d \times N}$ be a matrix consisting of hidden vectors $[h_1, \ldots, h_N]$ that the LSTM produced, where d is the size of hidden layers and N is the length of the given sentence. Furthermore, v_a represents the embedding of aspect and $e_N \in \mathbb{R}^N$ is a vector of 1s. The attention mechanism will produce an attention weight vector α and a weighted hidden

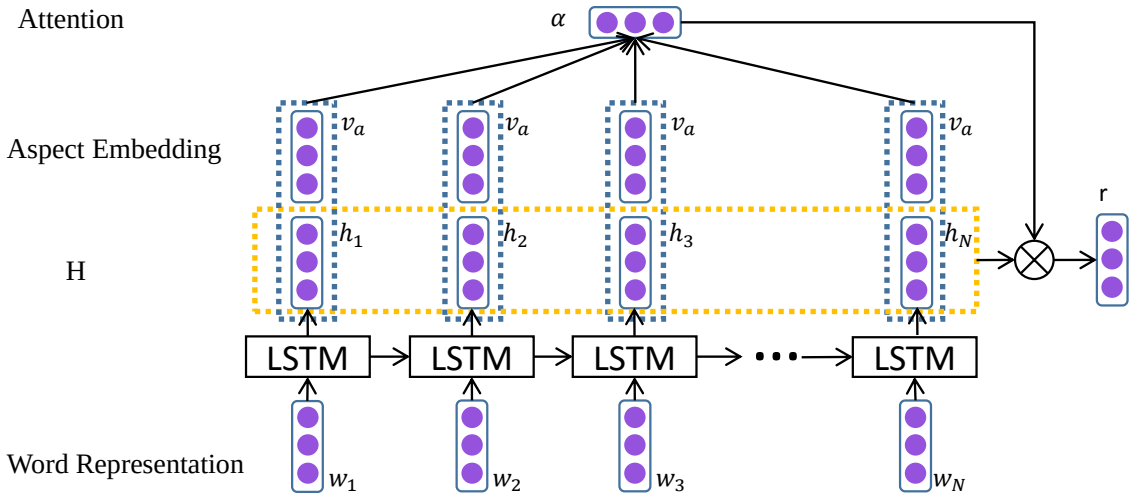


Figure 2: The Architecture of Attention-based LSTM. The aspect embeddings have been used to decide the attention weights along with the sentence representations. $\{w_1, w_2, \ldots, w_N\}$ represent the word vector in a sentence whose length is N. v_a represents the aspect embedding. α is the attention weight. $\{h_1, h_2, \ldots, h_N\}$ is the hidden vector.

representation r.

$$M = tanh(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix}) \quad (7)$$

$$\alpha = softmax(w^T M) \quad (8)$$

$$r = H\alpha^T \quad (9)$$

where, $M \in \mathbb{R}^{(d+d_a)\times N}$, $\alpha \in \mathbb{R}^N$, $r \in \mathbb{R}^d$. $W_h \in \mathbb{R}^{d\times d}$, $W_v \in \mathbb{R}^{d_a \times d_a}$ and $w \in \mathbb{R}^{d+d_a}$ are projection parameters. α is a vector consisting of attention weights and r is a weighted representation of sentence with given aspect. The operator in 7 (a circle with a multiplication sign inside, OP for short here) means: $v_a \otimes e_N = [v; v; \ldots; v]$, that is, the operator repeatedly concatenates v for N times, where e_N is a column vector with N 1s. $W_v v_a \otimes e_N$ is repeating the linearly transformed v_a as many times as there are words in sentence.

The final sentence representation is given by:

$$h^* = tanh(W_p r + W_x h_N) \quad (10)$$

where, $h^* \in \mathbb{R}^d$, W_p and W_x are projection parameters to be learned during training. We find that this works practically better if we add $W_x h_N$ into the final representation of the sentence, which is inspired by (Rocktäschel et al., 2015).

The attention mechanism allows the model to capture the most important part of a sentence when different aspects are considered.

h^* is considered as the feature representation of a sentence given an input aspect. We add a linear layer to convert sentence vector to e, which is a real-valued vector with the length equal to class number $|C|$. Then, a $softmax$ layer is followed to transform e to conditional probability distribution.

$$y = softmax(W_s h^* + b_s) \quad (11)$$

where W_s and b_s are the parameters for $softmax$ layer.

3.4 Attention-based LSTM with Aspect Embedding (ATAE-LSTM)

The way of using aspect information in AE-LSTM is letting aspect embedding play a role in computing the attention weight. In order to better take advantage of aspect information, we append the input aspect embedding into each word input vector. The structure of this model is illustrated in 3. In this way, the output hidden representations $(h_1, h_2, ..., h_N)$ can have the information from the input aspect (v_a). Therefore, in the following step that compute the attention weights, the inter-

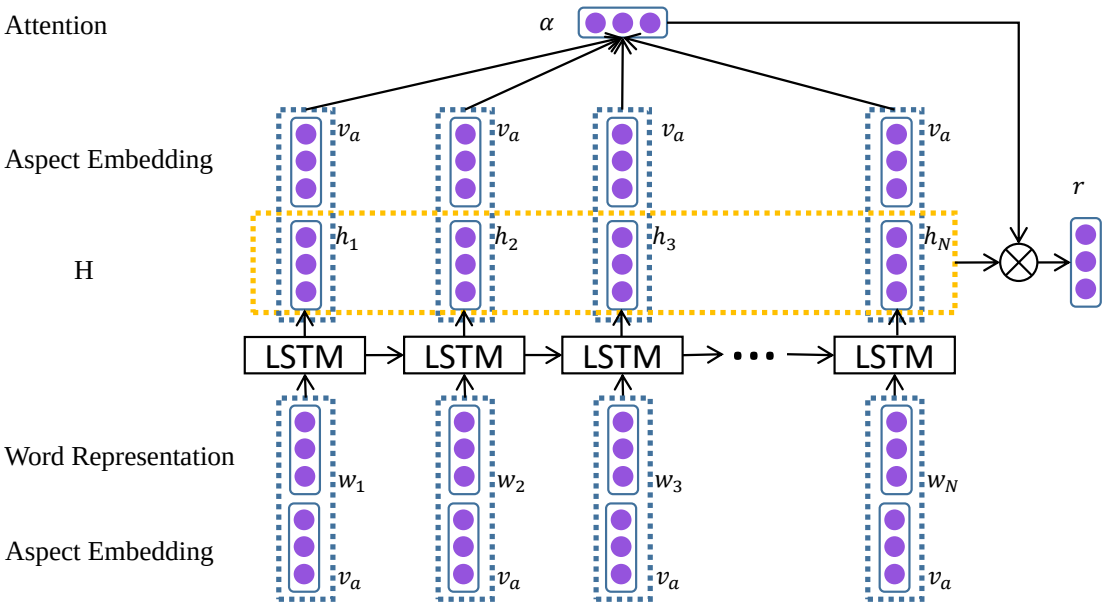


Figure 3: The Architecture of Attention-based LSTM with Aspect Embedding. The aspect embeddings have been take as input along with the word embeddings. $\{w_1, w_2, \ldots, w_N\}$ represent the word vector in a sentence whose length is N. v_a represents the aspect embedding. α is the attention weight. $\{h_1, h_2, \ldots, h_N\}$ is the hidden vector.

dependence between words and the input aspect can be modeled.

3.5 Model Training

The model can be trained in an end-to-end way by backpropagation, where the objective function (loss function) is the cross-entropy loss. Let y be the target distribution for sentence, $\hat{y}$ be the predicted sentiment distribution. The goal of training is to minimize the cross-entropy error between y and $\hat{y}$ for all sentences.

$$loss = -\sum_i \sum_j y_i^j log\hat{y}_i^j + \lambda||\theta||^2 \qquad (12)$$

where i is the index of sentence, j is the index of class. Our classification is three way. λ is the L_2 - regularization term. θ is the parameter set.

Similar to standard LSTM, the parameter set is $\{W_i, b_i, W_f, b_f, W_o, b_o, W_c, b_c, W_s, b_s\}$. Furthermore, word embeddings are the parameters too. Note that the dimension of W_i, W_f, W_o, W_c changes along with different models. If the aspect embeddings are added into the input of the LSTM cell unit, the dimension of W_i, W_f, W_o, W_c will be enlarged correspondingly. Additional parameters are listed as follows:

AT-LSTM: The aspect embedding A is added into the set of parameters naturally. In addition, W_h, W_v, W_p, W_x, w are the parameters of attention. Therefore, the additional parameter set of AT-LSTM is $\{A, W_h, W_v, W_p, W_x, w\}$.

AE-LSTM: The parameters include the aspect embedding A. Besides, the dimension of W_i, W_f, W_o, W_c will be expanded since the aspect vector is concatenated. Therefore, the additional parameter set consists of $\{A\}$.

ATAE-LSTM: The parameter set consists of $\{A, W_h, W_v, W_p, W_x, w\}$. Additionally, the dimension of W_i, W_f, W_o, W_c will be expanded with the concatenation of aspect embedding.

The word embedding and aspect embedding are optimized during training. The percentage of out-of-vocabulary words is about 5%, and they are randomly initialized from $U(-\epsilon, \epsilon)$, where $\epsilon = 0.01$.

In our experiments, we use AdaGrad (Duchi et al., 2011) as our optimization method, which has

improved the robustness of SGD on large scale learning task remarkably in a distributed environment (Dean et al., 2012). AdaGrad adapts the learning rate to the parameters, performing larger updates for infrequent parameters and smaller updates for frequent parameters.

4 Experiment

We apply the proposed model to aspect-level sentiment classification. In our experiments, all word vectors are initialized by Glove[1] (Pennington et al., 2014). The word embedding vectors are pre-trained on an unlabeled corpus whose size is about 840 billion. The other parameters are initialized by sampling from a uniform distribution $U(-\epsilon, \epsilon)$. The dimension of word vectors, aspect embeddings and the size of hidden layer are 300. The length of attention weights is the same as the length of sentence. Theano (Bastien et al., 2012) is used for implementing our neural network models. We trained all models with a batch size of 25 examples, and a momentum of 0.9, L_2-regularization weight of 0.001 and initial learning rate of 0.01 for AdaGrad.

4.1 Dataset

We experiment on the dataset of SemEval 2014 Task 4[2] (Pontiki et al., 2014). The dataset consists of customers reviews. Each review contains a list of aspects and corresponding polarities. Our aim is to identify the aspect polarity of a sentence with the corresponding aspect. The statistics is presented in Table 1.

4.2 Task Definition

Aspect-level Classification Given a set of pre-identified aspects, this task is to determine the polarity of each aspect. For example, given a sentence, "*The restaurant was too expensive.*", there is an aspect *price* whose polarity is negative. The set of aspects is {*food*, *price*, *service*, *ambience*, *anecdotes/miscellaneous*}. In the dataset of SemEval 2014 Task 4, there is only restaurants data that has aspect-specific polarities. Table 2 illustrates the comparative results.

[1]Pre-trained word vectors of Glove can be obtained from `http://nlp.stanford.edu/projects/glove/`

[2]The introduction about SemEval 2014 can be obtained from `http://alt.qcri.org/semeval2014/`

Asp.	Positive		Negative		Neural	
	Train	Test	Train	Test	Train	Test
Fo.	867	302	209	69	90	31
Pr.	179	51	115	28	10	1
Se.	324	101	218	63	20	3
Am.	263	76	98	21	23	8
An.	546	127	199	41	357	51
Total	2179	657	839	222	500	94

Table 1: Aspects distribution per sentiment class. {*Fo., Pr., Se, Am., An.*} refer to {*food, price, service, ambience, anecdotes/miscellaneous*}. "*Asp.*" refers to aspect.

Models	Three-way	Pos./Neg.
LSTM	82.0	88.3
TD-LSTM	82.6	89.1
TC-LSTM	81.9	89.2
AE-LSTM	82.5	88.9
AT-LSTM	83.1	89.6
ATAE-LSTM	**84.0**	**89.9**

Table 2: Accuracy on aspect level polarity classification about restaurants. *Three-way* stands for 3-class prediction. *Pos./Neg.* indicates binary prediction where ignoring all neutral instances. Best scores are in bold.

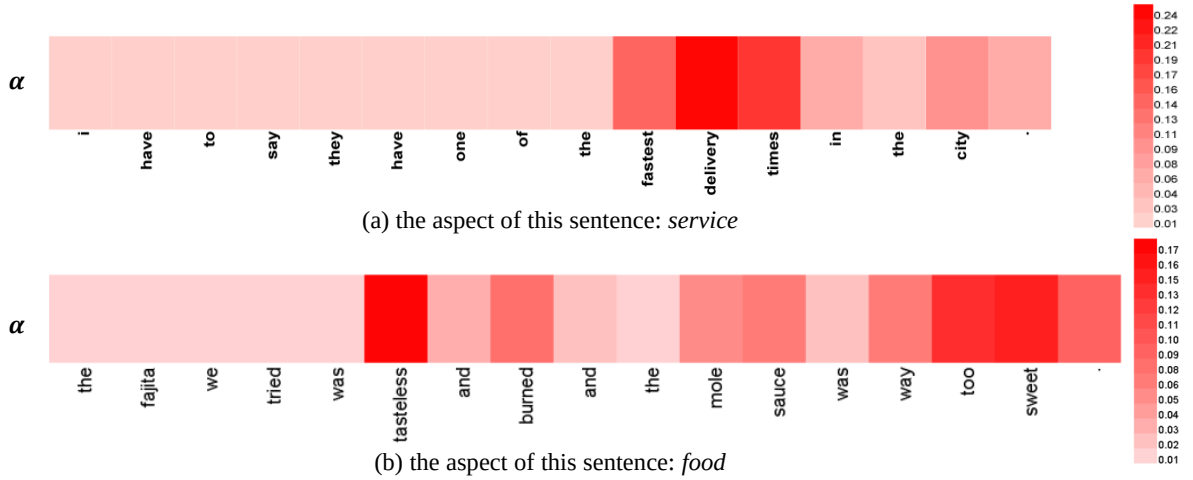
Aspect-Term-level Classification For a given set of aspects term within a sentence, this task is to determine whether the polarity of each aspect term is positive, negative or neutral. We conduct experiments on the dataset of SemEval 2014 Task 4. In the sentences of both restaurant and laptop datasets, there are the location and sentiment polarity for each occurrence of an aspect term. For example, there is an aspect term *fajitas* whose polarity is negative in sentence "*I loved their fajitas.*".

Experiments results are shown in Table 3 and Table 4. Similar to the experiment on aspect-level classification, our models achieve state-of-the-art performance.

4.3 Comparison with baseline methods

We compare our model with several baselines, including LSTM, TD-LSTM, and TC-LSTM.

LSTM: Standard LSTM cannot capture any aspect information in sentence, so it must get the same



(a) the aspect of this sentence: *service*

(b) the aspect of this sentence: *food*

Figure 4: Attention Visualizations. The aspects of (a) and (b) are *service* and *food* respectively. The color depth expresses the importance degree of the weight in attention vector α. From (a), attention can detect the important words from the whole sentence dynamically even though multi-semantic phrase such as "*fastest delivery times*" which can be used in other areas. From (b), attention can know multi-keypoints if more than one keypoint existing.

Models	Three-way	Pos./Neg.
LSTM	74.3	-
TD-LSTM	75.6	-
AE-LSTM	76.6	89.6
ATAE-LSTM	**77.2**	**90.9**

Table 3: Accuracy on aspect term polarity classification about restaurants. *Three-way* stands for 3-class prediction. *Pos./Neg.* indicates binary prediction where ignoring all neutral instances. Best scores are in bold.

Models	Three-way	Pos./Neg.
LSTM	66.5	-
TD-LSTM	68.1	-
AE-LSTM	**68.9**	87.4
ATAE-LSTM	68.7	**87.6**

Table 4: Accuracy on aspect term polarity classification about laptops. *Three-way* stands for 3-class prediction. *Pos./Neg.* indicates binary prediction where ignoring all neutral instances. Best scores are in bold.

sentiment polarity although given different aspects. Since it cannot take advantage of the aspect information, not surprisingly the model has worst performance.

TD-LSTM: TD-LSTM can improve the performance of sentiment classifier by treating an aspect as a target. Since there is no attention mechanism in TD-LSTM, it cannot "know" which words are important for a given aspect.

TC-LSTM: TC-LSTM extended TD-LSTM by incorporating a target into the representation of a sentence. It is worth noting that TC-LSTM performs worse than LSTM and TD-LSTM in Table 2. TC-LSTM added target representations, which was obtained from word vectors, into the input of the LSTM cell unit.

In our models, we embed aspects into another vector space. The embedding vector of aspects can be learned well in the process of training. ATAE-LSTM not only addresses the shortcoming of the unconformity between word vectors and aspect embeddings, but also can capture the most important information in response to a given aspect. In addition, ATAE-LSTM can capture the important and different parts of a sentence when given different aspects.

4.4 Qualitative Analysis

It is enlightening to analyze which words decide the sentiment polarity of the sentence given an aspect. We can obtain the attention weight α in Equation 8 and visualize the attention weights accordingly.

Figure 4 shows the representation of how attention focuses on words with the influence of a given aspect. We use a visualization tool Heml (Deng

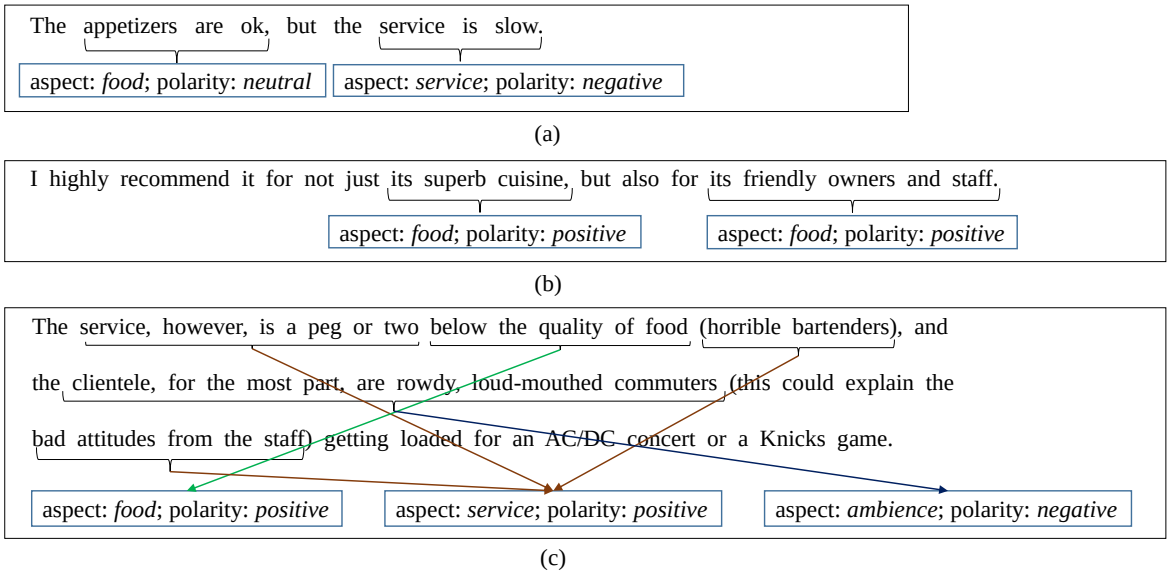


Figure 5: Examples of classification. (a) is an instance with different aspects. (b) represents that our model can focus on where the keypoints are and not disturbed by the privative word *not*. (c) stands for long and complicated sentences. Our model can obtain correct sentiment polarity.

et al., 2014) to visualize the sentences. The color depth indicates the importance degree of the weight in attention vector α, the darker the more important. The sentences in Figure 4 are "*I have to say they have one of the fastest delivery times in the city* ." and "*The fajita we tried was tasteless and burned and the mole sauce was way too sweet.*". The corresponding aspects are *service* and *food* respectively. Obviously attention can get the important parts from the whole sentence dynamically. In Figure 4 (a), "*fastest delivery times*" is a multi-word phrase, but our attention-based model can detect such phrases if *service* can is the input aspect. Besides, the attention can detect multiple keywords if more than one keyword is existing. In Figure 4 (b), *tastless* and *too sweet* are both detected.

4.5 Case Study

As we demonstrated, our models obtain the state-of-the-art performance. In this section, we will further show the advantages of our proposals through some typical examples.

In Figure 5, we list some examples from the test set which have typical characteristics and cannot be inferred by LSTM. In sentence (a), "*The appetizers are ok, but the service is slow.*", there are two aspects *food* and *service*. Our model can discriminate different sentiment polarities with different aspects. In sentence (b), "*I highly recommend it for not just its superb cuisine, but also for its friendly owners and staff.*", there is a negation word *not*. Our model can obtain correct polarity, not affected by the negation word who doesn't represent negation here. In the last instance (c), "*The service, however, is a peg or two below the quality of food (horrible bartenders), and the clientele, for the most part, are rowdy, loud-mouthed commuters (this could explain the bad attitudes from the staff) getting loaded for an AC/DC concert or a Knicks game.*", the sentence has a long and complicated structure so that existing parser may hardly obtain correct parsing trees. Hence, tree-based neural network models are difficult to predict polarity correctly. While our attention-based LSTM can work well in those sentences with the help of attention mechanism and aspect embedding.

5 Conclusion and Future Work

In this paper, we have proposed attention-based LSTMs for aspect-level sentiment classification. The key idea of these proposals are to learn aspect

embeddings and let aspects participate in computing attention weights. Our proposed models can concentrate on different parts of a sentence when different aspects are given so that they are more competitive for aspect-level classification. Experiments show that our proposed models, AE-LSTM and ATAE-LSTM, obtain superior performance over the baseline models.

Though the proposals have shown potentials for aspect-level sentiment analysis, different aspects are input separately. As future work, an interesting and possible direction would be to model more than one aspect simultaneously with the attention mechanism.

Acknowledgments

This work was partly supported by the National Basic Research Program (973 Program) under grant No.2012CB316301/2013CB329403, the National Science Foundation of China under grant No.61272227/61332007, and the Beijing Higher Education Young Elite Teacher Project. The work was also supported by Tsinghua University Beijing Samsung Telecom R&D Center Joint Laboratory for Intelligent Media Computing.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. 2012. Large scale distributed deep networks. In Advances in Neural Information Processing Systems, pages 1223–1231.

Wankun Deng, Yongbo Wang, Zexian Liu, Han Cheng, and Yu Xue. 2014. Hemi: a toolkit for illustrating heatmaps. PloS one, 9(11):e111988.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In ACL (2), pages 49–54.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, 12:2121–2159.

David Golub and Xiaodong He. 2016. Character-level question answering with attention. arXiv preprint arXiv:1604.00727.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1684–1692.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In EMNLP-CoNLL, pages 1075–1083.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Bing Liu. 2012. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1):1–167.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In INTERSPEECH, volume 2, page 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In Advances in Neural Information Processing Systems, pages 2204–2212.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. arXiv preprint arXiv:1308.6242.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In EMNLP, volume 4, pages 412–418.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture, pages 70–77. ACM.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), 12:1532–1543.

Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning sentiment lexicons in spanish. In LREC, volume 12, page 73.

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.

Qiao Qian, Bo Tian, Minlie Huang, Yang Liu, Xuan Zhu, and Xiaoyan Zhu. 2015. Learning tag embeddings and tag-specific composition functions in recursive neural network. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, volume 1, pages 1365–1374.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pages 675–682. Association for Computational Linguistics.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 151–161. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, volume 1631, page 1642. Citeseer.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015a. Target-dependent sentiment classification with long short term memory. arXiv preprint arXiv:1512.01100.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1422–1432.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193.