Aspect Aware Learning for Aspect Category Sentiment Analysis

PEISONG ZHU, ZHUANG CHEN, HAOJIE ZHENG, and TIEYUN QIAN, Wuhan University

Aspect category sentiment analysis (ACSA) is an underexploited subtask in aspect level sentiment analysis. It aims to identify the sentiment of predefined aspect categories. The main challenge in ACSA comes from the fact that the aspect category may not occur in the sentence in most of the cases. For example, the review "*they have delicious sandwiches*" positively talks about the aspect category "*food*" in an implicit manner.

In this article, we propose a novel aspect aware learning (AAL) framework for ACSA tasks. Our key idea is to exploit the interaction between the aspect category and the contents under the guidance of both sentiment polarity and predefined categories. To this end, we design a two-way memory network for integrating AAL into the framework of sentiment classification. We further present two algorithms to incorporate the potential impacts of aspect categories. One is to capture the correlations between aspect terms and the aspect category like *"sandwiches"* and *"food."* The other is to recognize the aspect category for sentiment representations like *"food"* for *"delicious."* We conduct extensive experiments on four SemEval datasets. The results reveal the essential role of AAL in ACSA by achieving the state-of-the-art performance.

CCS Concepts: • **Information systems** → **Data mining**;

Additional Key Words and Phrases: Aspect category sentiment analysis, aspect aware learning, memory network

ACM Reference format:
Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect Aware Learning for Aspect Category Sentiment Analysis. *ACM Trans. Knowl. Discov. Data* 13, 6, Article 55 (October 2019), 21 pages.
https://doi.org/10.1145/3350487

1 INTRODUCTION

Sentiment analysis [9, 16, 22, 23] is a fundamental task in natural language processing. Aspect level sentiment analysis [10, 20, 28] is a fine-grained task, which aims at identifying the sentiment polarity of a specific target. Due to the ability of providing thorough and detailed results, it is arousing more and more researchers' interests.

There are two subtasks in aspect level sentiment classification. The first is to identify the sentiment of predefined aspect categories, called as ACSA (aspect category sentiment analysis) for short. The second is to identify the sentiment of given terms that occur in the sentence, called as

The work described in this article has been supported in part by the NSFC projects (61572376, 91646206), and the 111 project(B07037).

Authors' addresses: P. Zhu, Z. Chen, H. Zheng, and T. Qian (corresponding author), 16 Luojiashan Road, School of Computer Science, Wuhan University, Wuhan, Hubei, China, 430072; email: {zhups24, zhchen18, zhenghaojie, qty}@whu.edu.cn.

1556-4681/2019/10-ART55 $15.00
https://doi.org/10.1145/3350487

ATSA (aspect term sentiment analysis). Let us take the review "*Not only they have delicious sandwiches, soup, pizza etc, but the waiters are so courteous*!" as an example. ACSA aims to infer the sentiment polarities for a set of predefined aspect categories like "*food*" and "*service*," while ATSA aims to infer the sentiment polarities for given terms "*sandwiches,*" "*soup,*" "*pizza,*" and "*waiters.*"

ATSA has been extensively studied since an early age [10, 20, 21, 28, 30]. The success of deep learning methods further gives a tremendous boost to this field [1, 3, 7, 14, 17, 18, 31, 35, 37, 38]. On the other hand, ACSA is surprisingly underexploited despite its wide applications like review summarization and recommendation. Only limited research [33, 34, 39–42] has been done toward ACSA.

The main challenge in ACSA lies in the fact that aspect categories usually do not occur in the sentence, and it is hard for a model to locate the exact position of the aspect category, not to mention discerning its contexts. In the previous example, we have two positive sentiment polarity for two implicit aspect categories "*food*" and "*service*," respectively. In this case, the attention based mechanism [39] may selectively concentrate on "*delicious*" for "*service*" and "*courteous*" for "*food*," or focus on both for "*service*" or "*food.*" Obviously, such attentions are questionable because "*delicious*" can only be used to modify "*food.*" The problem may be caused by the following two issues. One is that we cannot locate two aspect categories "*food*" and "*service*" in the sentence. The other is that the model will not get the error feedback by interactively selecting "*delicious*" and "*courteous*" since both of them reflect positive polarity.

Both issues arise from the lack of guidance of aspect category (in the following, we use "aspect" and "aspect category" interchangeably unless explicitly pointed) information during the learning of the representation of words or the entire sentence. To address this problem, we propose a novel aspect aware learning (AAL) framework for ACSA task. Our intuition is that besides the sentiment polarity, the predefined aspect information can be used to supervise the learning procedure. Specifically, we design a two-way neural network: one for sentiment classification, and the other for supervised AAL. Our main contribution lies in AAL. In order to make use of aspect information, we first present an approach to capture the connections between aspect terms and the aspect, e.g., "*sandwiches, soup, pizza*" and "*food.*" We also propose to recognize the aspect for sentiment representations. For example, we hope to identify the aspect "*food*" given the sentiment specified by "*delicious.*" We conduct extensive experiments on four benchmark datasets from SemEval 2014–2016. The results prove that our model achieves the state-of-the-art performance on ACSA.

The rest of this article is structured as follows. In Section 2, we present the related work. In Section 3, we introduce our proposed model. In Section 4, we show the experimental evaluation. In Section 5, we present the deep analysis results. We conclude the article in Section 6.

2 RELATED WORK

We review the literature in the area of aspect level sentiment analysis, and we are particularly interested in recent advance of utilizing the deep learning technique.

2.1 Aspect Level Sentiment Analysis

Aspect level sentiment classification is a subtask of sentiment analysis in the field of NLP. Traditional methods [10, 21, 30] usually use machine learning algorithms to build sentiment classifier with carefully extracted features, which take massive time and resources. Recent years have witnessed the boom of deep learning methods in aspect level sentiment classification tasks. The studies are dominated by methods for ATSA that integrate target information [3, 14, 17, 18, 24, 32, 35, 37, 38]. The key difference between ACSA and ATSA is that term targets explicitly occur in the sentence while the aspect categories that reflect high-level semantics rarely exist in the sentence. Hence, such kinds of methods for ATSA cannot be applied to ACSA tasks.

Several pioneering methods [33, 34, 39–42] are developed towards ACSA recently. Wang et al. [39] proposed an attention-based LSTM method by concentrating on different parts of a sentence to different aspects. Xue and Li [40] adopted gated convolutional networks (GCN) to extract aspect-specific sentiment information. Tay et al. [33, 34] designed a dyadic memory network and aspect fusion LSTM to model interactions between aspect and context. Yang et al. [41] took context, entity, and aspect information into consideration via memory network and attention mechanism. Zhu and Qian [42] adopted two memory networks to capture the important context words for sentiment classification and to implicitly convert aspects and terms to each other.

Overall, current studies in ACSA mainly make use of the emerging network architecture like LSTM, GCN, and memory network to focus on the informative contexts towards the aspect embedding. None of them employs the aspect to supervise the learning process. We also use memory network for implementation. However, we differentiate our work with existing studies in that we integrate the AAL into the framework of sentiment classification. The incorporation of supervised aspect information helps recognize the important sentiment and aspect terms, and consequently improves the performance.

2.2 Deep Learning Framework in Aspect Level Sentiment Analysis

Our study is inspired by the recent advances in neural networks, including the unsupervised aspect extraction method in [6], the deep multi-task learning framework in [15] for aspect and opinion extraction tasks, the end-to-end deep memory network for attitude identification and polarity classification task in [13], and the neural attention networks for stance classification task in [4].

He et al. [6] proposed an attention-based model for unsupervised aspect extraction. The main intuition is to utilize the attention mechanism to focus more on aspect-related words during the learning of aspect embeddings. Li and Lam [15] proposed a deep multi-task learning framework where two LSTMs equipped with extended memories and neural memory operations are designed for jointly handling the extraction tasks of aspects and opinions via memory interactions. Li et al. [13] proposed an end-to-end deep memory network to model the interaction of target entities in attitude identification task and polarity classification task. Target-Specific Neural Attention Networks is a model proposed by [4] for stance classification task, which is similar to aspect level sentiment classification. This model learns target-augmented embeddings for text and uses attention mechanism to extract target-specific parts in text to improve classification performance.

We distinguish our work with those using memory networks [32, 33] in that our goal is to capture the relatedness between term and aspect to improve the sentiment classification performance and we design an auxiliary memory to this end. In contrast, existing studies only focus on the relatedness between context words the aspect/term using a sentiment memory, which is a single component in our model. We will show in the experiment that the proposed AAL component is critical in enhancing the sentiment analysis performance.

3 OUR PROPOSED AAL MODEL

This section we present our AAL model. We first present the overall architecture of our model. We then introduce its three main components.

3.1 An Overview

The architecture of AAL model is shown in Figure 1. It has three components, i.e., the network input component on the bottom left, the AAL component on the upper left, and the sentiment classification component on the right.

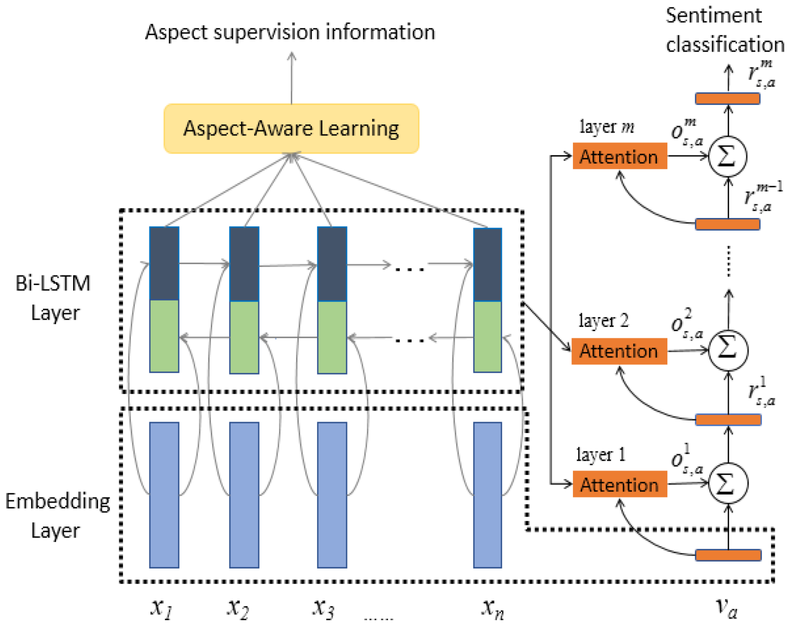


Fig. 1. Architecture of AAL Model.

The network input component consists of an embedding layer and a Bi-LSTM layer (two dashed boxes). The hidden representation of the sentence generated by the input component, along with the aspect embedding, is fed into sentiment memory (the three layer in red on the right) to extract the sentiment representation for the aspects. The hidden representation is also used as the input to the AAL component, which integrates the aspect information into the hidden representation so as to capture the correlation between the words in the sentence and the predefined aspects.

3.2 Network Input Component

The network input consists of an embedding layer and a Bi-LSTM layer. The former maps each word into a low-dimensional continuous vector, and the latter captures the forward and backward information of a sentence to a specific word.

Embedding layer: To apply deep-learning method to text data, we map each word into a low-dimensional continuous vector, which is known as word embedding [19, 25]. Specifically, let $\mathbb{L} \in \mathbb{R}^{d \times |V|}$ be an embedding matrix made up of all the word embeddings, where d is the dimensionality of word embedding, and $|V|$ is vocabulary size. Given a sentence $S = \{w_1, w_2, \ldots, w_n\}$ containing n words, we use a lookup layer to get the embedding $x_i \in \mathbb{R}^d$ of word w_i, which is a column in the embedding matrix $\mathbb{L}$. In addition, we use the matrix $\mathbb{A} \in \mathbb{R}^{da \times |A|}$ to represent all aspects, where da is the dimensionality of aspect embedding, and $|A|$ is the number of predefined aspects. Given a predefined aspect a, we use a lookup layer to get its embedding $v_a \in \mathbb{R}^{da}$.

Bi-LSTM layer: Long short term memory (LSTM) [8] has been widely used in various NLP tasks. At each time step t, LSTM has a set of parameters $\{i_t, f_t, o_t, c_t, h_t\}$, where i, f, o, c, and h denote the input gate, forget gate, output gate, cell state, and hidden state, respectively. These gates adaptively remember input vector, forget previous history and calculate o_t and c_t as follows:

$$\begin{aligned} i_t &= \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i), \\ f_t &= \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f), \\ o_t &= \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o), \\ \hat{c_t} &= tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c), \\ c_t &= f_t \square c_{t-1} + i_t \square \hat{c_t}, \end{aligned} \tag{1}$$

where σ is the sigmoid function, x_t is the word embedding for current word w_t, and W, U, and b denote the weight matrix and bias, respectively. $\square$ denotes element-wise product. The current hidden state h_t can be computed from the current cell state c_t and output o_t as follows:

$$h_t = o_t \square tanh(c_t), \tag{2}$$

We adopt a bi-direction LSTM to capture the forward and backward information in a sentence. Each word w_i will get two hidden states $\overleftarrow{h_i}$ and $\overrightarrow{h_i}$ and its final representation is the concatenation of these two states, i.e., $h_i = \overleftarrow{h_i} \oplus \overrightarrow{h_i}$. In summary, when a sentence S containing n words is input into the Bi-LSTM layer, it will be encoded as a hidden matrix $\mathbb{H} \in \mathbb{R}^{n \times 2d_h}$ consisting of n columns. Each column h_i is a $2d_h$ vector denoting the hidden representation for the ith word in S, and d_h is the dimensionality for one hidden state. We concisely define this process as follows:

$$H = Bi - LSTM(X, \Theta_{LSTM}), \tag{3}$$

where $\mathbb{X} \in \mathbb{R}^{n \times d}$ is the word embedding matrix for the input sentence, and Θ_{LSTM} denotes the network parameters.

3.3 Sentiment Classification Component

The goal of ACSA is to identify the sentiment polarity of predefined aspects. The memory network can yield improved results on question answering [29] and ATSA task [32]. Hence, we also adopt a memory network, called sentiment memory, for sentiment classification. It accepts an aspect category as the query and computes interactions between the query and context via attention mechanism. Stacking multiple hops in memory network can uncover more complicated linguistic evidence, which are useful for sentiment prediction.

Sentiment memory: The sentiment memory consists of multiple layers. We use the superscript to denote the layer number below. The sentiment memory cells are initialized with the hidden matrix $\mathbb{H} \in \mathbb{R}^{n \times 2d_h}$, which is got from the Bi-LSTM layer for the sentence S. In the first layer, the sentiment memory takes the hidden representation $h_i \in \mathbb{R}^{2d_h}$ and the aspect embedding v_a as input, and generates a weighted hidden vector $o^1_{s,a} \in \mathbb{R}^{2d_h}$, which is computed as a weighted sum of all hidden representations via attention mechanism:

$$\begin{aligned} o^1_{s,a} &= \sum_{i=1}^{n} h_i \beta^1_{i,a}, \\ \beta^1_{i,a} &= softmax(tanh(W_s[h_i, v_a] + b_s)), \end{aligned} \tag{4}$$

where $\beta^1_{i,a}$ is the attention weight vector calculated by the relatedness between h_i and v_a using a forward neural network, W_s and b_s are the weight matrix and bias, respectively. We concatenate $o^1_{s,a}$ and v_a as the output $r^1_{s,a}$ of the first layer:

$$r^1_{s,a} = o^1_{s,a} + v_a, \tag{5}$$

$r^1_{s,a}$ is then used as the input of the second layer to replace v_a in Equations (4) and (5), and we concatenate $o^2_{s,a}$ and $r^1_{s,a}$ to get the output $r^2_{s,a}$ of the second layer. In the similar way, we can stack multiple (five in our setting) layers. We summarize the stacking process as follows:

$$\begin{aligned} o^m_{s,a} &= \sum_{i=1}^{n} h_i \beta^m_{i,a}, \\ \beta^m_{i,a} &= softmax\Big(tanh\Big(W_s[h_i, r^{m-1}_{s,a}] + b_s\Big)\Big), \\ r^m_{s,a} &= o^m_{s,a} + r^{m-1}_{s,a}. \end{aligned} \tag{6}$$

Sentiment classification: The output $r^m_{s,a}$ of the last layer in sentiment memory is used for sentiment classification. The conditional probability of sentiment polarity c given the sentence S and the aspect a is defined as follows:

$$\hat{p}(c|S,a) = softmax\Big(W_c r^m_{s,a} + b_c\Big), \tag{7}$$

where W_c and b_c denote the weight matrix and bias, respectively. The objective of sentiment classification is to minimize the cross entropy loss between the predicted and the ground truth sentiment distribution. We use L_{senti} to denote this loss function and define it as follows:

$$L_{senti} = -\sum_{(S,a)\in T}\sum_{c\in C} p(c|S,a)\cdot \hat{p}(c|S,a), \tag{8}$$

where S is the sentence in training corpus T and C the set of sentiment polarity. $p(c|S,a)$ is the ground truth probability of sentiment category c given the sentence S and aspect a.

3.4 Aspect Aware Learning Component

We now present our two algorithms, i.e., AAL-Lex and AAL-SS, for AAL component.

3.4.1 AAL-Lex algorithm. Sentiment lexicons have been shown helpful for sentiment analysis [9, 20, 35, 36]. However, no previous work has explored aspect lexicons jointly with neural networks to improve aspect level sentiment classification. In this subsection, we propose our aspect lexicon based AAL algorithm (AAL-Lex).

We first construct an aspect lexicon that reflects the correlation between each word and the aspects like "*pizza*" and "*food*." Inspired by existing studies on building sentiment lexicons [2, 5, 12], we propose to use the point-wise mutual information (PMI) to construct the aspect lexicon. The PMI between the word w and aspect a is defined as follows:

$$PMI(w,a) = log\frac{N(w,a)\times N(S)}{N(w)\times N(a)}, \tag{9}$$

where $N(S)$, $N(w)$, $N(a)$, and $N(w,a)$ denote the number of sentences in the corpus, that containing the word w, that annotated with the aspect a, and that containing w, and annotated with a, respectively. Note that we remove the words whose frequency is lower than five. This is because these words will have high PMI scores, which are often unreliable due to the low frequency. We also set 0 PMI value for stop words since they usually contain no realistic meanings and act as noise. We apply the softmax function on PMI to get the conditional probability of aspect a given the word w.

$$p(a|w) = \frac{exp(PMI(w,a))}{\sum_{j=1}^{|A|} exp(PMI(w,a_j))}, \tag{10}$$

The aspect lexicon is then used as the ground-truth aspect information to supervise the training of aspect classification. More specifically, we do aspect classification for each word in the sentence.

The goal is to integrate the aspect information into the hidden representation. To this end, we compute the predicted probability of aspect a using the hidden representation h for word w:

$$\hat{p}(a|w) = softmax(W_a h + b_a), \tag{11}$$

The objective of aspect classification is to minimize the cross entropy loss between the predicted and the ground truth aspect distribution. We use L_{asplex} to denote this loss function and define it as:

$$L_{asplex} = -\sum_{S \in T} \sum_{w \in S} \sum_{a \in A} p(a|w) \cdot log\hat{p}(a|w), \tag{12}$$

where S is the sentence in training corpus T and A is the aspect set.

Finally, we combine the loss function in Equation (8) for sentiment classification and that in Equation (12) for aspect classification into our AAL-Lex algorithm. The overall loss function is defined as follows:

$$L_{aallex} = (1 - \gamma_1)L_{senti} + \gamma_1 L_{asplex}, \tag{13}$$

where γ_1 is the hyper-parameter that balances the weight of sentiment classification and that of aspect classification in AAL-Lex.

3.4.2 AAL-SS Algorithm. AAL-Lex algorithm identifies the aspect category for each word in the sentence, and its performance highly relies on the corpus from which the lexicon is constructed. For example, if a food lexicon is built from restaurants and is about main dishes like "*beefsteak*," the model cannot be successfully applied to those about fast foods like "*hamburger*." In this subsection, we present our sentence level supervision based AAL (AAL-SS) algorithm. The basic idea is to identify the aspect category for the entire sentence based on the sentiment representation of the sentence. On one hand, the sentiment words are related with aspects. For example, "*delicious*" and "*courteous*" can be used to describe "*food*" and "*service*" and they are not interchangeable. On the other hand, the sentiment words will be more general than aspect terms. For example, we can use "*delicious*" for most kinds of foods no matter the food is "*beefsteak*" or "*hamburger*."

The input to AAL-SS is the hidden representation $\mathbb{H}$ introduced in the previous section. We first use the attention mechanism to get the sentiment representation of the sentence as follows:

$$\begin{aligned} M &= tanh(W_h H + b_h), \\ \alpha &= softmax(u^T M), r_a = H\alpha^T, \end{aligned} \tag{14}$$

where W_h and b_h denote the weight matrix and bias, respectively. $u \in \mathbb{R}^{2dh \times 1}$ is a randomly initialized contextual vector for sentiment. It will be updated during the training process. r_a is the sentiment representation for the sentence.

We then associate the sentiment representation with the aspect. We do this by computing the predicted probability of aspect a using the sentiment representation r_a for the sentence S as follows:

$$\hat{p}(a|S) = \sigma(W_a r_a + b_a), \tag{15}$$

where W_a and b_a denote the weight matrix and bias, σ is the sigmoid function. The objective of aspect classification is to minimize the cross entropy loss between the predicted and the ground truth aspect distribution.

Note that in Equation (15) we use sigmoid instead of softmax in Equation (11) for AAL-Lex. This is because the basic unit in AAL-Lex is word. Equation (11) uses Softmax to characterize comparative relations of a word to aspect category. For example, in "great food but the service was dreadful," the correlation of (dreadful, service) is stronger than that of (dreadful, food). AAL-SS is at sentence level and does not have such relations. Equation (15) uses Sigmoid to output the probability for each of multiple aspect categories in the sentence.

We use L_{aspss} to denote this loss function and define it as follows:

$$L_{aspss} = -\sum_{S \in T} \sum_{a \in A} y(S,a) \cdot \hat{p}(a|S), \tag{16}$$

where $y(S,a)$ is an indicator function with 1 denoting the sentence S describing the aspect a and 0 otherwise.

Finally, we combine the loss function in Equation (8) for sentiment classification and that in Equation (16) for aspect classification into our AAL-SS algorithm. The overall loss function is defined as follows:

$$L_{aalss} = (1-\gamma_2)L_{senti} + \gamma_2 L_{aspss}, \tag{17}$$

where γ_2 is the hyper-parameter, which balances the weight of sentiment classification and that of aspect classification in AAL-SS.

3.4.3 Model Training. Given the loss functions defined in Equations (13) and (17), our AAL model can be trained in an end-to-end way by back propagation. Specifically, we use Tensorflow 1.2 for implementing our neural network models. We adopt Adam [11] as our optimization method and set the learning rate as 0.01. The batch size is set to 25 examples. We train the model for 100 batches (one epoch) with early stopping if the performance on the development set does not improve after 10 epochs. The model with the smallest loss on the development set is then used to classify the test data.

4 EXPERIMENTS

We apply our proposed AAL model to aspect category sentiment classification to evaluate its effectiveness.

4.1 Settings

4.1.1 Datasets. We use four datasets including Restaurant-2014 from SemEval 2014 Task 4 [28], Laptop-2015 and Restaurant-2015 from SemEval 2015 Task 12 [27] and Restaurant-2016 datasets from SemEval 2016 Task 5 [26]. Each review in these datasets is annotated with a sentiment polarity towards a given aspect category. The detailed aspect categories in each dataset are summarized in Table 1. Note that Laptop-2015 contains the entity information that will be used in CEA baseline [41].

There are four categories of sentiment label, i.e.,"*Positive,*" "*Negative,*" "*Neutral,*" and "*Conflict.*" Since the number of samples in "*conflict*" is tiny, we remove this category as previous work does [33, 34, 39, 41, 42]. We use the official training/testing split in our experiments. Furthermore, following the previous work in [33, 34], we randomly sample 500, 300, 250, and 350 samples as the development set to fine-tune the hyper-parameters on four datasets, respectively. The data statistics are shown in Table 1.

4.1.2 Compared Methods. To comprehensively evaluate the performance of our proposed AAL model, we compare it with the existing state-of-the-art baselines for ACSA.

- *AE-LSTM* [39] takes into account aspect information by learning an embedding vector for each aspect. This is the first work to use aspect embedding.
- *ATAE-LSTM* [39] is an extension of AE-LSTM. It uses attention mechanism to 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.

Table 1. Data Statistics

Data	Aspect	Positive		Negative		Neutral	
		Train	Test	Train	Test	Train	Test
Restaurant-2014	food	867	302	209	69	90	31
	price	179	51	115	28	10	1
	service	324	101	218	63	20	3
	ambience	263	76	98	21	23	8
	anecdotes/miscellaneous	546	127	199	41	357	51
	Total	2179	657	839	222	500	94
Laptop-2015	general	366	187	154	71	6	15
	price	40	35	25	5	22	17
	quality	110	55	266	60	10	4
	operation_performance	154	82	111	76	9	5
	usability	106	32	42	26	10	11
	design_features	142	65	63	39	32	16
	portability	36	5	8	2	0	1
	connectivity	17	6	15	15	0	3
	miscellaneous	70	43	33	21	12	5
	Total	1041	513	717	315	101	77
Restaurant-2015	service#general	153	40	95	110	7	5
	food#quality	328	153	95	59	13	10
	restaurant#general	217	93	47	50	5	2
	drinks#style_options	23	4	1	2	0	0
	drinks#prices	11	2	4	3	0	0
	restaurant#prices	29	6	14	28	5	1
	ambience#general	127	45	22	17	8	6
	food#style_options	56	19	23	15	5	4
	restaurant#miscellaneous	41	19	17	12	3	7
	food#prices	25	8	22	19	1	2
	drinks#quality	31	7	1	4	1	1
	location#general	17	4	2	0	1	4
	food#general	0	0	1	0	0	0
	Total	1058	400	344	319	49	42
Restaurant-2016	restaurant#general	312	107	100	34	8	1
	service#general	194	66	206	70	12	7
	food#quality	480	186	153	24	23	12
	food#style_options	76	25	41	14	9	8
	drinks#style_options	27	11	3	1	0	0
	drinks#prices	13	0	7	3	0	0
	restaurant#prices	34	6	40	13	6	2
	restaurant#miscellaneous	57	16	27	13	13	4
	ambience#general	171	52	34	1	15	3
	food#prices	36	6	44	13	1	3
	location#general	21	11	1	0	6	2
	drinks#quality	39	20	5	1	2	0
	Total	1460	506	661	187	95	42

- *Tensor DyMemNN* [33] is one variation of the dyadic memory networks that incorporate interactions between aspect and document into the memory network by using tensor products.
- *Holo DyMemNN* [33] is similar to Tensor DyMemNN. It uses holographic compositions to replace tensor products.
- *CEA* [41] consists of context, entity, and aspect memory to combine these types of information. Since the Restaurant dataset does not label the entity, we set the entity as a zero vector as that in [41].
- *GCAE* [40] is a model based on convolutional neural networks and gating mechanisms, which can selectively extract aspect-specific sentiment information for sentiment prediction.
- *DAuM* [42] adopts two memory networks to simultaneously learn features of aspects and terms.
- *AF-LSTM (CONV)* [34] adopts circular convolution to model the similarity between aspect and words and incorporates this within a differentiable neural attention framework.
- *AF-LSTM (CORR)* [34] is the other implementation of AF-LSTM, which uses circular correlation instead of circular convolution.
- *AAL-No* is a simplified variant of our proposed AAL framework, which removes the AAL component. Please note that AAL-SS and AAL-Lex will become identical after removing this component, and thus we only have one variant for these two algorithms.

The baselines are optimized using the parameters and optimization methods reported in their original papers.

4.1.3 Evaluation Protocol. We present our evaluation protocol including the metrics and settings.

Metrics: We adopt the standard *Accuracy* [34, 39–42] and the macro-averaged *Precision*, *Recall*, and *F-score* (macro-F1) [33, 41, 42] as the evaluation metrics to show a more thorough comparison with previous studies.

Settings: The dimension of word embedding is set to 300, which is same as those in baselines. The word vectors are pre-trained by GloVe [25] and are fixed when training the model. Other parameters like aspect embedding (dimensionality 300), weight matrices and biases are initialized by sampling from a uniform distribution U(0.01, 0.01), and are optimized during the training process. The remaining hyper-parameters are fine-tuned on the development set. γ_1, γ_2 are set to {0.3, 0.3, 0.1, 0.2} and {0.4, 0.4, 0.3, 0.3} on {Restaurant-2014, Laptop-2015, Restaurant-2015, Restaurant-2016}, respectively. The number of hops m is set to 5. We will investigate the effects of these parameters later.

4.2 Comparison to Other Methods

We compare our proposed model with other methods, and show the results in Tables 2 and 3.

It is clear that our AAL-SS algorithm achieves the best performance on all four datasets. It gets much better Accuracy and F-score than all baselines and variants. Meanwhile, our AAL-LEX algorithm also achieves a competitive performance compared with baselines. The simplified version AAL-No, which contains no AAL components, performs inferior to these two approaches.

CEA, GCAE, and DAuM have relatively superior performance among the baselines. CEA adopts two interaction layers to characterize the relation between aspect categories and context words, and it can utilize the entity information on Laptop-2015 dataset. The result of CEA shows that the sentiment information extraction could benefit from the enhanced interaction. GCAE performs relative steady on all datasets by combining the gating mechanism with CNN feature extractor,

Table 2. Experimental Results on Restaurant-2014 and Laptop-2015 Datasets

Method	Restaurant-2014				Laptop-2015			
	Acc.	Pre.	Rec.	F-score	Acc.	Pre.	Rec.	F-score
AE-LSTM	81.40	71.78	65.22	67.63	74.39	56.94	55.66	54.73
ATAE-LSTM	82.32	73.95	70.45	71.96	74.50	56.42	55.77	55.60
Tensor DyMemNN	80.99	73.22	65.04	68.10	75.66	60.88	55.12	53.47
Holo DyMemNN	80.37	72.00	67.76	69.62	76.08	66.57	55.37	53.05
CEA	82.94	73.23	69.01	70.81	74.50	59.35	55.82	56.51
GCAE	81.09	69.93	65.88	67.61	75.03	60.79	59.80	59.96
DAuM	81.50	75.51	64.66	67.92	76.19	50.00	54.63	52.21
AF-LSTM (CORR)	82.01	74.83	71.24	72.01	76.19	49.93	54.87	52.26
AF-LSTM (CONV)	82.22	72.41	**74.63**	73.32	76.29	50.04	54.77	52.29
AAL-No	83.63	71.25	72.22	71.67	75.17	60.34	56.26	57.61
AAL-LEX	84.17	75.25	73.96	74.57	75.87	62.18	58.14	59.25
AAL-SS	**85.61**	**78.03**	73.71	**75.54**	**78.29**	**67.75**	**59.82**	**60.00**

Note: Best scores are in bold.

Table 3. Experimental Results on Restaurant-2015 and Restaurant-2016 Datasets

Method	Restaurant-2015				Restaurant-2016			
	Acc.	Pre.	Rec.	F-score	Acc.	Pre.	Rec.	F-score
AE-LSTM	76.08	52.39	52.52	51.42	80.95	62.62	58.74	58.25
ATAE-LSTM	76.87	51.13	54.34	52.61	81.09	71.09	55.65	58.86
Tensor DyMemNN	78.58	59.59	55.58	55.08	82.99	76.01	62.14	63.45
Holo DyMemNN	77.92	52.16	54.39	53.08	82.59	69.53	58.23	59.93
CEA	78.19	66.73	58.00	59.36	82.72	77.66	60.26	62.04
GCAE	77.53	60.93	57.42	58.22	82.59	72.26	61.32	61.86
DAuM	78.98	67.87	57.98	59.01	81.36	66.58	64.09	63.95
AF-LSTM (CORR)	77.40	51.55	54.25	52.84	83.27	71.17	61.31	62.20
AF-LSTM (CONV)	76.61	51.84	53.18	52.01	81.90	69.07	59.14	61.29
AAL-No	76.74	63.55	56.64	57.81	82.31	65.88	64.27	64.62
AAL-LEX	77.92	63.59	57.32	58.14	84.08	**80.00**	64.06	65.56
AAL-SS	**79.11**	**70.44**	**60.61**	**62.80**	**84.35**	77.68	**64.64**	**67.14**

Note: Best scores are in bold.

which demonstrates the usability of CNN-based model in ACSA tasks. DAuM considers the dependency of aspect terms and aspect categories. It achieves better performance on Restaurant-2015 and Restaurant-2016 datasets than other datasets, the reason might be that the predefined aspect categories on these two datasets are relatively adequate and distinguishable.

AF-LSTM and Holo DyMemNN adopt the circular correlation and circular convolution operations to capture the interaction between aspect categories and context words, while Tensor DyMemNN is armed with neural tensor composition to drive the memory selection process in memory network. These models are not stable on different datasets, which shows the additional interaction methods do not generalize well. AE-LSTM and ATAE-LSTM achieve relatively poor performance compared with other baselines, since they only contain shallow LSTM and attention layers, which are not enough to extract deep semantic meanings.

It is clear that though most of baselines pay attention to the interaction or similarity between aspect categories and context words, their performances tend to fluctuate without the guidance of explicit aspect information. In contrast, our AAL-Lex and AAL-SS models enhance the interaction between the aspect learning and the sentiment classification task, and thus outperform all the baselines.

4.3 With or Without Aspect Aware Learning

We would highlight the difference between our incomplete AAL variant without aspect supervision (AAL-No), and two complete AAL models with aspect supervision (AAL-Lex and AAL-SS). It can be seen from Tables 2 and 3 that, without the aspect supervision information, the performance of AAL-No drops dramatically. For example, the overall F-score of AAL-No on four datasets is 71.67%, 57.61%, 57.81%, and 64.62%, respectively, significantly worse than that of AAL-Lex (74.57%, 59.25%, 58.14%, and 65.56%) and AAL-SS (75.54%, 60.00%, 62.80%, and 67.14%). This clearly demonstrates that our model benefits a lot from the AAL component.

Indeed, when the supervision component is removed from the AAL architecture, the model degrades into a simple multi-layer memory network for aspect category classification. Its performance is close to that of DyMemNN, which is also based on memory network. This clearly distinguishes our work from those use memory network as basic architecture.

4.4 Comparison Between AAL-Lex and AAL-SS

Another interesting finding from Tables 2 and 3 is that between our two algorithms, AAL-SS performs better than AAL-Lex. The reason may be that AAL-Lex relies on the recognized words related to a category. If the corpus cannot ensure the coverage and stability of the aspect lexicon, AAL-Lex may fail to work. We use the following example to show why AAL-SS outperforms AAL-Lex.

Case 1: *also, the chick peas with shrimp (appetizer) is divine*

In case 1, the word "*divine*" is positive in the training "*food*" category. However, since it occurs only once in the sentence "*It was divine melts in your mouth.*" in training set, it is not included in the lexicon and AAL-Lex makes a wrong prediction. In contrast, AAL-SS gives a correct answer as it not only learns the positive representation for "*divine*" but also finds the connection between "*divine*" and "*food.*"

While AAL-SS works well in many cases, sometimes it is not as good as AAL-Lex as we show in the following example.

Case 2: *so yes, it is ridiculously fast.*

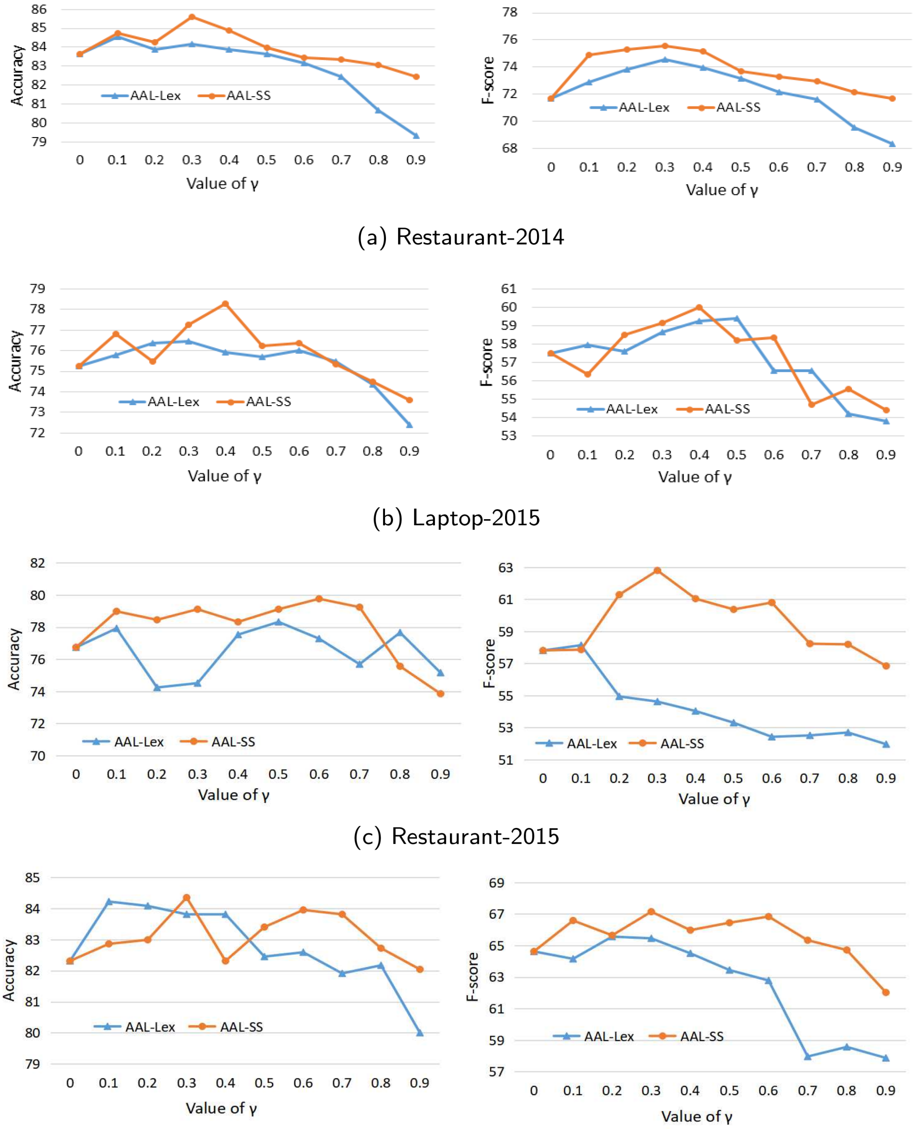
In case 2, both AAL-Lex and AAL-SS emphasize on "*fast*," which is a positive word for computer performance. However, AAL-SS does not assign as large attention on "*fast*" as AAL-Lex does. Moreover, AAL-SS highlights "*ridiculously*" as a negation word. This results in the wrong and correct prediction of AAL-SS and AAL-Lex, respectively.

Overall, AAL-SS is flexible and generalizable to unseen words. In practice, AAL-Lex can be applied to specific domains with fixed vocabulary while AAL-SS is appropriate for open domains.

4.5 Effects of Balance Factor

The balance factor γ (including γ_1 for AAL-Lex and γ_2 for AAL-SS) determines how important the AAL is in the entire model. To evaluate the effects of the balance factor, we change γ from 0.0 to 0.9 stepped by 0.1. The results on four datasets are shown in Figure 2(a) and (d), respectively.

We take the curves on Restaurant-2014 as the example to illustrate the effects of gamma, while similar phenomena could be found on other datasets. In Figure 2(a), we can observe an upward trend for the performance of our AAL model when $\gamma < 0.3$. When γ is set to 0.0, it becomes the



(a) Restaurant-2014

(b) Laptop-2015

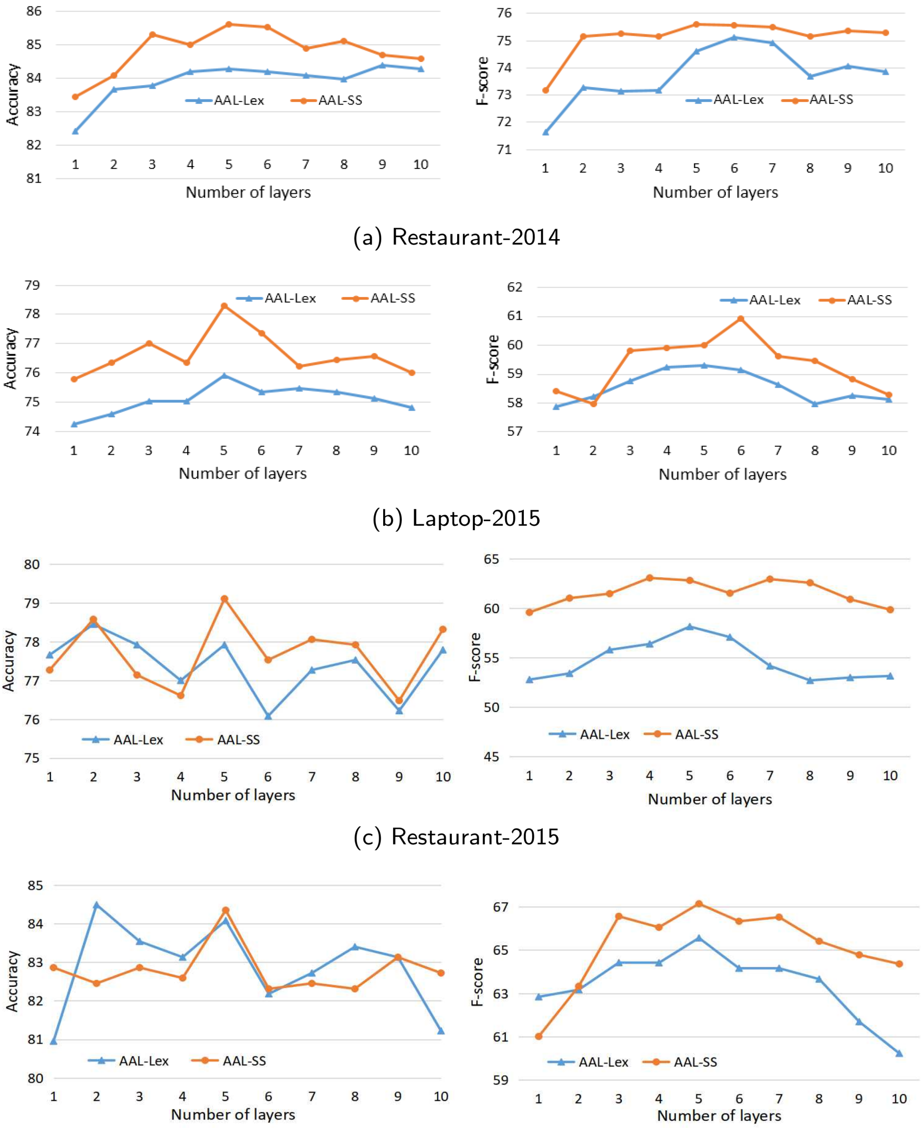
(c) Restaurant-2015

(d) Restaurant-2016

Fig. 2. Effects of γ.

AAL-No variant (see Tables 2 and 3) since the model does not contain any aspect information and is reduced to the basic sentiment classification model. Increasing γ introduces the AAL into the model, and achieves the best performance with the γ value 0.3. This proves that our AAL method can add aspect supervision information for the word or the sentence representations; thus, the sentiment classification can precisely extract the sentiment representation for the aspect.

We also notice a decreasing trend when γ is greater than 0.3. This is reasonable because a large γ will enforce the model to learn for aspect classification and neglect sentiment information. As a result, the model learnt without a proper objective cannot obtain good performance for sentiment classification. For practical use, γ can be tuned on the development set as we do in this article.



(a) Restaurant-2014

(b) Laptop-2015

(c) Restaurant-2015

(d) Restaurant-2016

Fig. 3. Effects of layer number.

4.6 Effects of Layer Number

In this section, we examine the effects of layer number. The classification results in terms of accuracy and F-score by changing the layer number are shown in Figure 3(a)–(d), respectively.

As we illustrate in the previous section, we adopt sentiment memory in sentiment classification component, which may consist of one or more computational layers (hops). The intuition of utilizing multiple hops is that more abstractive evidence could be uncovered based on previously extracted evidence. Existing studies show that using multiple layers could yield improved results on question answering [29] and term sentiment classification [32]. Hence, we investigate the

Table 4. Aspect Lexicon on Restaurant-2014

Aspect Category	Top-10 aspect terms with the highest PMI
food	tasted, melted, burnt, roast, grilled, oily, shredded, martini, crispy, egg
price	inexpensive, reasonably, $, free, pricey, priced, cost, prices, price, reasonable
service	server, servers, smile, courteous, ignored, greeted, helpful, manager, phone, asked,
ambience	outdoor, paris, sleek, scene, music, cramped, laid-back, romantic, cozy, comfortable,
anecdote/miscellaneous	anniversary, stumbled, reading, based, month, somewhere, celebrate, opened, katz, located

Table 5. Aspect Lexicon on Laptop-2015

Aspect Category	Top-10 aspect terms with the highest PMI
general	gift, worst, loved, products, pleased, recommended, friends, hate, happy, stars
price	expensive, shipping, paid, price, cost, spent, $, worth, deal, money, fixed
quality	crap, defective, year, customer, told, loud, waited, piece, hot, quiet
operation performance	freezes, hrs, seconds, loads, flawlesslym, freaking, runs, applications, stopped, blue
usability	navigate, curve, learning, friendly, learn, switch, ease, window, user, os
design features	offers, sleek, feature, ram, sized, place, design, features, allow, ports
portability	travel, portability, carry, portable, meets, durable, student, fit, business, sit
connectivity	wifi, network, wireless, ethernet, port, connection, connect, plugged, stay, camera
miscellaneous	media, word, handle, facebook, microsoft, basic, games, gaming, stuff, trial

effects of multiple layers on our aspect sentiment classification. We change our model by increasing the number of layers from 1 to 10 stepped by 1 with the fixed γ.

When the number of layers is less than three, it is clear that using more layers can help find abstractive representation and improve the performance. The best performances are achieved when the model contains five or six layers. However, the performance does not always get enhanced with the increasing number of layers. Once the number of layers exceeds a certain threshold, the model would be difficult to train and less generalizable due to the excessive computing complexity.

4.7 Aspect Lexicon

The aspect lexicon is critical for AAL-Lex algorithm. Since there is no evaluation metric for measuring the quality of the lexicon, we show the top-10 aspect terms on four datasets with the highest PMI scores to each aspect category in Tables 4–7, respectively.

Table 6. Aspect Lexicon on Restaurant-2015

Aspect Category	Top-10 aspect terms with the highest PMI
service#general	asked, waitress, rude, waiter, group, attentive, fast, problem, prompt, n't
food#quality	bagels, spicy, bland, authentic, salad, pad, tasting, sweet, cooked, perfection
restaurant#general	favorite, pleasantly, find, wrong, loved, gem, will, restaurants, Mizu, return
drinks#style_options	extensive, interesting, glass, list, house, priced, sake, wine, wines, simple
drinks#prices	n't, music, extensive, love, cheap, wines, average, priced, wine, selection
restaurant#prices	high, inexpensive, set, money, poor, cramped, house, felt, reasonable, prices
ambience#general	affordable, soho, village, avenue, romantic, cozy, music, garden, upper, beer
food#style_options	limited, sliced, choices, ingredients, slice, lot, appetizer, portions, rolls, order
restaurant#miscellaneous	totally, waitstaff, japanese, spot, specials, owner, friends, dinner, business, night
food#prices	return, bite, chinatown, expensive, expected, overpriced, dim, sum, visit, better
drinks#quality	fine, interesting, decent, selection, fun, wines, drinks, ordered, glass, hot
location#general	location, view, neighborhood, notch, saul, ambiance, live, suan, trip, avenue
food#general	N/A (no sample in training set)

We take Table 4 as the example for illustrating the aspect lexicon on the restaurant domain. It can be seen from Table 4 that almost all terms are highly correlated with the four aspects "*food*," "*price*," "*service*," and "*ambience*." For example, the top-1 terms related to these four aspects are "*tasted*," "*inexpensive*," "*server*," and "*outdoor*," respectively. The only exception is "*paris*," which looks like irrelevant with "*ambience*." However, in the real reviews, this term is actually used to modify the atmosphere in the sentence like "*I felt as though I were eating in paris*." The terms in "anecdotes/miscellaneous" are not strongly correlated with their aspect. The reason may be that the semantics of sentences in this category are usually ambiguous and sometimes obscure. Originally, we believed the words like "*delicious*" should be aspect dependent. However, they do not rank high in the lexicon. This phenomenon might be caused by the co-occurrence with multiple aspects in the same sentence like "*delicious appetizer and friendly staff*," which lowers the conditional probability of $p(food|delicious)$.

Table 5 shows the aspect lexicon on the laptop domain. For example, when talking about the aspect category of "*connectivity*," the top-5 words are "*wifi*," "*network*," "*wireless*," "*ethernet*," and "*port*," which are exactly about the connectivity of a computer. The top-3 words in "*operation performance*" are "*freeze*," "*hrs*," and "*seconds*." While the last two words are used to measure the speed or frequency, the first word denotes that the system hangs or stops.

Table 7. Aspect Lexicon on Restaurant-2016

Aspect Category	Top-10 aspect terms with the highest PMI
restaurant#general	warn, mileau, casa, femme, gem, fare, favorite, la, return, loved
service#general	manager, slow, waitress, rude, attentive, walked, waiter, maitre, finally, customers
food#quality	bland, authentic, bagels, oily, dogs, dry, enjoyed, exceptional, caviar, pork
food#style_options	limited, looked, huge, range, spinach, variety, choices, size, portions, bistro
drinks#style_options	extensive, list, interesting, reasonably, priced, wine, guest, bottles, husband, choices
drinks#prices	bottles, bottle, n't, music, pay, piece, told, water, cheap, love
restaurant#prices	paying, aggressive, pricey, range, expensive, inexpensive, set, bark, high, poor
restaurant#miscellaneous	occasion, waitstaff, busy, reservations, specials, open, packed, totally, problem, seated
ambience#general	soho, relax, relaxed, trendy, cheaply, manhattanite, pretentious, clientele, roosevelt, vibe
food#prices	overpriced, management, fixe, waiters, expected, party, mediocre, prix, fair, expensive
location#general	view, hidden, location, notch, ambiance, Suan, live, fact, spectacular, pay
drinks#quality	glass, homey, fairly, selection, wines, interesting, completely, fine, listed, waited

In general, PMI can be a good indicator to choose aspect related words. However, it has the limitation that it is dependent on the training corpus. How to expand the lexicon is critical to the performance. We leave this as our future work.

5 DEEP ANALYSIS

In order to see how various methods differ on complicated cases, we take the following five reviews as examples. The first three cases contain one single aspect and the next two contains two aspects. Note these reviews are correctly classified by our AAL-SS (and/or AAL-Lex) model but wrongly labeled by all baselines.

Case 3: *Do n't be put off by another reviewer's extremely negative reviews!*

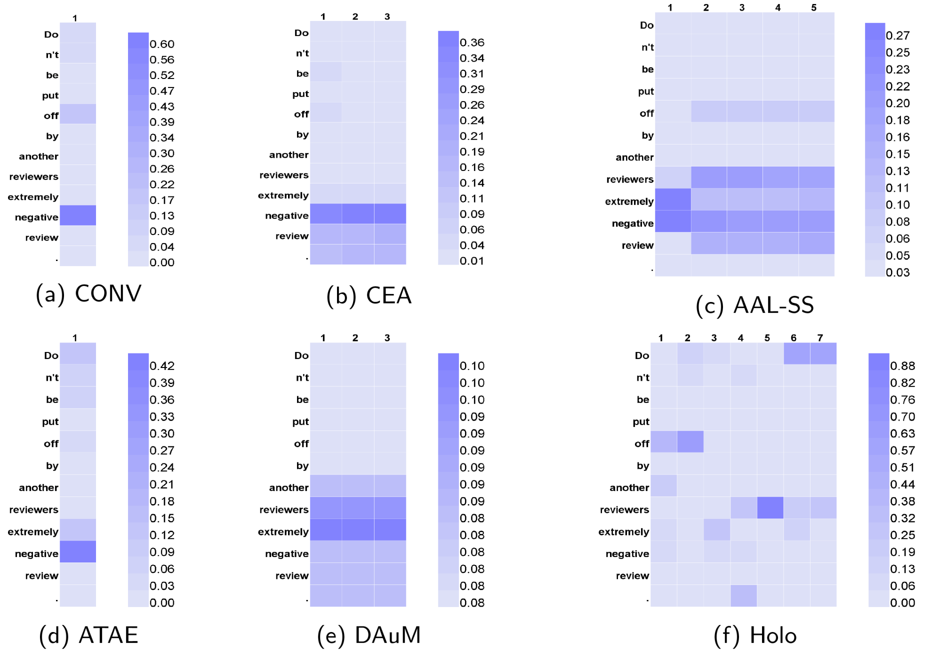
Case 4: *I wasn't disappointed in its performance.*

Case 5: *For someone who used to hate Indian food, Baluchi's has changed my mid.*

Case 6: *How can they survive serving mediocre food at exorbitant prices?!*

Case 7: *Although the restaurant itself is nice, I prefer not to go for the food.*

The first three cases contain a negation. The negative polarity is triggered by direct negatives "*n't*" and "*wasn't*" in case 3 and case 4, and also by an implicit word "*changed*" in case 5. All these make it difficult to discern the sentiment. To have a close look, we visualize the attentions by different methods for case 3 in Figure 4. We choose the ATAE, Holo, and CONV variant as the representative of two LSTM, DyMemNN, and AF-LSTM methods. Note we do not present the



(a) CONV (b) CEA (c) AAL-SS

(d) ATAE (e) DAuM (f) Holo

Fig. 4. Attention weights comparison for different methods on case 3. Aspect: *anec/mis*, sentiment: *positive*.

visualization for GCAE because it does not use attention mechanism. AAL-SS, CEA, DAuM, and Holo adopt multiple layers for memory network. CONV and ATAE can only use one layer. The layer columns show the word weights in each layer, indicated by color depth. The attention values for different methods vary from 0.00 to 0.88.

From Figure 4, it is clear most of models can correctly highlight the sentiment term "*negative.*" However, given the negation "*do n't be put off,*" it is hard for attention-based methods, such as CEA, ATAE, and DAuM, to capture the interactions between negation words and aspect words only by attention mechanism itself. Hence, these methods are unable to concentrate on the negation words properly.

On the other hand, AAL-SS, CONV, and Holo assign the large weight to "*off*" (the semantic end of negation). The auxiliary aspect classification task in AAL-SS helps the attention mechanism to find related context words. CONV and Holo exploit dyadic relations between aspect words and context words, so they also have the ability to recognize the negation words. Nevertheless, Holo is wrong since it does not recognize the sentiment word "*negative.*" The wrong decision of CONV may be due to that it only has one layer, and hence the negation of "*do n't be put off*" can not counteract the effect of "*negative.*" In contrast, our AAL-SS model assigns large weights to "*negative*" and its qualifier "*extremely*" with the supervision of the aspect "*review.*" Moreover, with the hlep of sentence-level supervision provided by aspect category classification task in AAL-SS, the negation words are related to the intensified aspect and sentiment words when training, and thus are properly specified in testing. All these help our model make the correct prediction.

We further present the results for case 6 and case 7 in Table 8.

We can see that all baselines are unable to discern the correct "*neutral*" sentiment for the "*food*" aspect in case 6. The reason may be that they are all misled by the negative sentiment for the "*price*" aspect. The wrong prediction on "*food*" in case 7 by the baselines are most probably caused by the negation. Our AAL-SS model emphasizes the conjunction and negation word "*although*" and "*not,*"

Table 8. Case Studies for Sentences with Two Aspects

Case		ATAE	Holo	CEA	GEAE	DAuM	CONV	AAL-SS
6	food	neg.	neg.	neg.	neg.	neg.	neg.	**neu.**
	price	neg.	neg.	neg.	neg.	neg.	neg.	**neg.**
7	ambience	pos.	pos.	neu.	pos.	pos.	pos.	**pos.**
	food	pos.	pos.	neu.	pos.	pos.	pos.	**neg.**

The boldface values denote that only AAL-SS makes correct prediction on both aspects in Case 6 and Case 7.

and relates the sentiment to its own aspect. Consequently, AAL-SS assigns correct polarity to both aspects.

6 CONCLUSION

We present a novel AAL model for aspect category sentiment classification. We implement it with a two-way neural network. The key advantage is to adopt the aspect information to supervise the learning of word representations in addition to the sentiment information. We design an aspect lexicon based and an aspect supervision based algorithm for AAL. Empirical results on four benchmark datasets from SemEval 2014–2016 verify that our AAL model achieves the state-of-the-art performance. For future work, we plan to model the deep interactions between the aspect and sentiments and simultaneously make use of aspect lexicon and aspect supervision.

REFERENCES

[1] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 452–461.

[2] Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16, 1(1990), 22–29.

[3] 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 *Annual Meeting of the Association for Computational Linguistics (ACL'14)*. 49–54.

[4] Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention. In *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 3988–3994.

[5] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 595–605.

[6] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL'17)*. 388–397.

[7] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 579–585.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[9] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[10] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* 151–160.

[11] Soo Min Kim and E. H. Crystal Hovy. 2007. Analyzing predictive opinions on the web. In *Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*. 1056–1064.

[12] O. Levy, Y. Goldberg, and I. Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3 (2015), 211–225.

[13] Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deep memory networks for attitude identification. In *10th ACM International Conference on Web Search and Data Mining (WSDM'17)*. 671–680.

[14] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 946–956.

[15] Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 2886–2892.

[16] Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Springer, Berlin.

[17] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 4068–4074.

[18] Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *AAAI Conference on Artificial Intelligence*.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR (Workshop Poster)*.

[20] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-canada: Building the state-of-the-art in sentiment analysis of tweets. In *International Workshop on Semantic Evaluation (SemEval'13)*.

[21] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 786–794.

[22] Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *2nd international conference on Knowledge capture (K-CAP'03)*. 70–77.

[23] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*. 79–86.

[24] Minlong Peng, Qi Zhang, Yugang Jiang, and Xuanjing Huang. 2018. Cross-domain sentiment classification with target domain specific information. In *Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 2505–2513.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.

[26] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval'16)*. 19–30.

[27] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval'15)*. 486–495.

[28] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect based sentiment analysis. *International Workshop on Semantic Evaluation (SemEval'14)*. 27–35.

[29] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NIPS'15)*. 2440–2448.

[30] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37, 2 (2011), 267–307.

[31] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *International Conference on Computational Linguistics (COLING'16)*. 3298–3307.

[32] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 214–224.

[33] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. Dyadic memory networks for aspect-based sentiment analysis. In *ACM International Conference on Information and Knowledge Management (CIKM'17)*. 107–116.

[34] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *AAAI Conference on Artificial Intelligence*.

[35] Duy-Tin Vo and Yue Zhang. 2015. Target dependent twitter sentiment classification with rich automatic features. In *International Joint Conferences on Artificial Intelligence (IJCAI'15)*. 1347–1353.

[36] Leyi Wang and Rui Xia. 2017. Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 502–510.

[37] Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 957–967.

[38] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 616–626.

[39] Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 606–615.

[40] Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 2514–2523.

[41] Jun Yang, Runqi Yang, Chong-Jun Wang, and Jun-Yuan Xie. 2018. Multi-entity aspect-based sentiment analysis with context, entity and aspect memory. In *AAAI Conference on Artificial Intelligence*.

[42] Peisong Zhu and Tieyun Qian. 2018. Enhanced aspect level sentiment classification with auxiliary memory. In *International Conference on Computational Linguistics (COLING'18)*. 1077–1087.

Received December 2018; revised July 2019; accepted July 2019