Better Queries for Aspect-Category Sentiment Classification

Yuncong Li [1, *], **Cunxiang Yin** [1, *], **Sheng-hua Zhong** [2, †], **Huiqiang Zhong** [1], **Jinchang Luo** [1], **Siqi Xu** [1] and **Xiaohui Wu** [1]
[1]Baidu Inc., Beijing, China
[2]College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen, China
[1]{liyuncong,yincunxiang,zhonghuiqiang,luojinchang,xusiqi01,
wuxiaohui02}@baidu.com
[2]csshzhong@szu.edu.cn

Abstract

Aspect-category sentiment classification (ACSC) aims to identify the sentiment polarities towards the aspect categories mentioned in a sentence. Because a sentence often mentions more than one aspect category and expresses different sentiment polarities to them, finding aspect category-related information from the sentence is the key challenge to accurately recognize the sentiment polarity. Most previous models take both sentence and aspect category as input and query aspect category-related information based on the aspect category. However, these models represent the aspect category as a context-independent vector called aspect embedding, which may not be effective enough as a query. In this paper, we propose two contextualized aspect category representations, Contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM). Specifically, we use the coarse aspect category-related information found by the aspect category detection task to generate CAV or CAM. Then the CAV or CAM as queries are used to search for fine-grained aspect category-related information like aspect embedding by aspect-category sentiment classification models. In experiments, we integrate the proposed CAV and CAM into several representative aspect embedding-based aspect-category sentiment classification models. Experimental results on the SemEval-2014 Restaurant Review dataset and the Multi-Aspect Multi-Sentiment dataset demonstrate the effectiveness of CAV and CAM.

1 Introduction

Sentiment analysis (Pang and Lee, 2008; Liu, 2012) is an important task in Natural Language Processing (NLP). It deals with the computational treatment of opinion, sentiment, and subjectivity in text. Aspect-based sentiment analysis (Pontiki et al., 2014; Pontiki et al., 2015; Pontiki et al., 2016) is a branch of sentiment analysis and aspect-category sentiment analysis (ACSA) is a subtask of it. In ACSA, there are a predefined set of aspect categories, and a predefined set of sentiment polarities. Given a sentence, the task aims to predict the aspect categories mentioned in the sentence and the corresponding sentiments. Therefore, ACSA contains two subtasks: aspect category detection (ACD) that detects aspect categories in a sentence and aspect-category sentiment classification (ACSC) that categorizes the sentiment polarities with respect to the detected aspect categories. Figure 1 shows an example, "Staffs are not that friendly, but the taste covers all.". ACD detects the sentence mentions two aspect categories: *service* and *food*, and ACSC predicts the sentiment polarities to them: negative and positive respectively. In this work, we focus on ACSC, while ACD as an auxiliary task is used to find coarse aspect category-related information for the ACSC task.

Because a sentence often mentions more than one aspect category and expresses different sentiment polarities to them, to accurately recognize the sentiment polarities, most previous models (Ruder et al., 2016; Wang et al., 2016; Cheng et al., 2017; Li et al., 2017; Tay et al., 2018; Xue and Li, 2018; Xing et al., 2019; Liang et al., 2019b; Jiang et al., 2019; Hu et al., 2019; Wang et al., 2019) take both sentence and aspect category as input and query aspect category-related information based on the aspect

[*] Equal contribution.

[†] Corresponding author.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1080-1089, Hainan, China, October 31 - November 1, 2020.

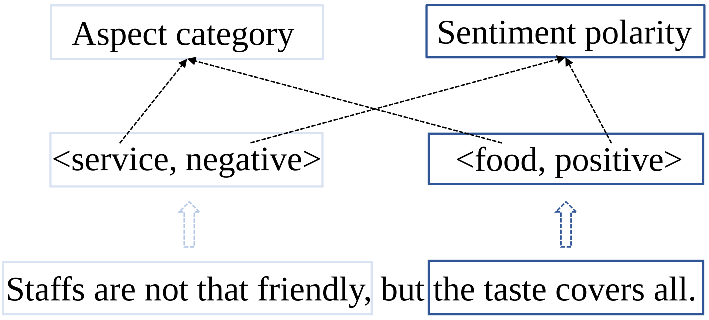


Figure 1: An example of aspect-category sentiment analysis.

category, then generate aspect category-specific representations for aspect-category sentiment classification. However, these models represent the aspect category as a context-independent vector called aspect embedding (AE). These models can be called aspect embedding-based models. Since aspect embedding only contains the global information of aspect category and loses the context-dependent information, it is semantically far away from the words in the sentence, and may not be effective enough as a query to search for aspect category-related information for the ACSC task. These models may be improved by replacing the aspect embedding with context-dependent aspect category representations.

The HiErarchical ATtention (HEAT) network (Cheng et al., 2017) used context-dependent aspect category representations to search for aspect category-related information for the ACSC task and obtained better performance. The context-dependent aspect category representations are generated by concatenating the aspect embedding and the aspect term representation in a sentence. An aspect term is a word or phrase that appears in the sentence explicitly indicating an aspect category. For the example in Figure 1, the aspect terms are “Staffs” and “taste” indicating aspect category *service* and *food* respectively. However, the HEAT network requires aspect term annotation information that the data for ACSC usually does not have. Moreover, the HEAT network ignores the situation where the aspect category is mentioned implicitly in sentences without any aspect term, making aspect category representations degenerate to context-independent representations in this situation.

In this paper, we propose two novel contextualized aspect category representations, Contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM). CAV or CAM contain context-dependent information even though there are no aspect terms in sentences, and aspect term annotation information is not required to generate them. Concretely, we use the coarse aspect category-related information found by the ACD task to generate CAV or CAM. Then CAV or CAM as queries are used to search for fine-grained aspect category-related information like aspect embedding by aspect-category sentiment classification models. Specifically, we first use an attention-based aspect category classifier to obtain the weights of the words in a sentence, which indicate the degree of correlation between the aspect categories and the words. Then, we get CAV by combining the weighted sum of the word representations with corresponding aspect embedding. That is to say, CAV contains two kinds of representations of an aspect category: context-independent representation and context-dependent representation, which capture global information and local information respectively. Since CAV may lose details of the words, we also propose an aspect category matrix representation, called Contextualized Aspect Matrix (CAM), which is a not-sum version of CAV.

In summary, the main contributions of our work can be summarized as follows:

- We propose two novel contextualized aspect category representations, Contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM). They include the global information and local information about the aspect category and are better queries to search for aspect category-related information for aspect category sentiment classification (ACSC). To the best of our knowledge, it is the first time to represent aspect category as matrix.

- We experiment with several representative aspect embedding-based models by replacing the aspect embedding with CAV or CAM. Experimental results on the SemEval-2014 Restaurant Review dataset and the Multi-Aspect Multi-Sentiment dataset demonstrate the effectiveness of CAV and CAM.

2 Related Work

In this section, we first present a brief review about aspect-category sentiment classification. Then, we show the related study on context-aware aspect embedding that is a kind of context-dependent aspect category representation for targeted aspect based sentiment analysis (TABSA).

2.1 Aspect-Category Sentiment Classification

Many models (Wang et al., 2016; Ruder et al., 2016; Cheng et al., 2017; Li et al., 2017; Tay et al., 2018; Schmitt et al., 2018; Xue and Li, 2018; Xing et al., 2019; Liang et al., 2019b; Jiang et al., 2019; Hu et al., 2019; Wang et al., 2019; Sun et al., 2019) have been proposed for the aspect-category sentiment classification (ACSC) task. Wang et al. (2016) proposed an attention-based LSTM network for aspect-level sentiment classification. Tay et al. (2018) introduced a word-aspect fusion attention layer to attend based on associative relationships between sentence words and aspect categories. Xue et al. (2018) proposed to extract sentiment features with convolutional neural networks and selectively output aspect category related features for classification with gating mechanisms. Xing et al. (2019) proposed a novel variant of LSTM, which incorporates aspect information into LSTM cells in the context modeling stage. Liang et al. (2019b) proposed a novel Aspect-Guided Deep Transition model, which utilizes the given aspect category to guide the sentence encoding from scratch. Jiang et al. (2019) proposed new capsule networks to model the complicated relationship between aspects and contexts. To force the orthogonality among aspect categories, Hu et al. (2019) proposed constrained attention networks (CAN) for multi-aspect sentiment analysis. To avoid error propagation, some joint models (Li et al., 2017; Schmitt et al., 2018; Wang et al., 2019) have been proposed, which perform aspect category detection (ACD) and aspect-category sentiment classification (ACSC) jointly. Li et al. (2017) proposed an end-to-end machine learning architecture, in which the ACD task and the ACSC task are interleaved by a deep memory network. Wang et al. (2019) proposed the aspect-level sentiment capsules model (AS-Capsules), which utilizes the correlation between aspect and sentiment through shared components including capsule embedding, shared encoders, and shared attentions. The capsule embedding is similar to the aspect embedding. All these models represented aspect category as context-independent representations, which may benefit from CAV or CAM.

Closely related to our method is the HiErarchical Attention (HEAT) network proposed by Cheng et al. (2017), in which an aspect attention extracts the aspect term information, and then a context-dependent aspect category representation generated based on the aspect term information is used to guide the sentiment attention to better allocate aspect-specific sentiment words of the text. However, extracting aspect term information requires additional aspect term annotation information. In addition, HEAT ignores the situation where the aspect category is mentioned implicitly in texts. There are also some models that don't rely on aspect embedding. Schmitt et al. (2018) also proposed a joint model, in which different aspect categories have different sentiment classifiers to generate aspect category-specific representations. Sun et al. (2019) constructed an auxiliary sentence from the aspect and converted ABSA to a sentence-pair classification task.

2.2 Context-aware Aspect Embedding

Context-aware aspect embedding is a kind of context-dependent aspect category representation (Liang et al., 2019a). Liang et al. (2019a) proposed an embedding refinement method to generate context-aware target embedding and aspect embedding for targeted aspect based sentiment analysis (TABSA) (Saeidi et al., 2016), which utilizes a sparse coefficient vector to adjust the embeddings of target and aspect from the context and yields the state-of-the-art performance in this task. However, their method relies on context-aware target embedding to generate aspect embedding, and can't be applied in the ACSC task directly.

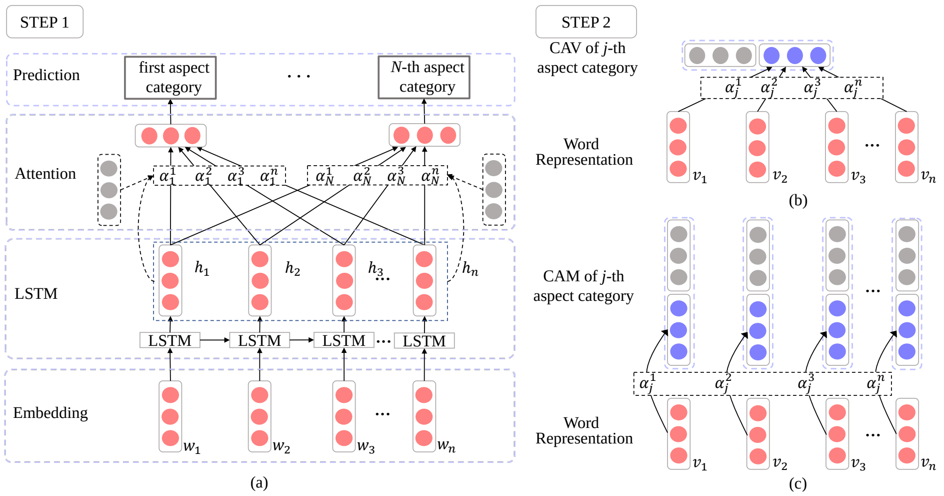

STEP 1
Prediction
first aspect category
. . .
N-th aspect category
Attention
α_1^1 α_1^2 α_1^3 α_1^n
α_N^1 α_N^2 α_N^3 α_N^n
LSTM
h_1 h_2 h_3 ... h_n
LSTM LSTM LSTM ... LSTM
Embedding
w_1 w_2 w_3 ... w_n
(a)
STEP 2
CAV of j-th aspect category
α_j^1 α_j^2 α_j^3 α_j^n
Word Representation
v_1 v_2 v_3 ... v_n
(b)
CAM of j-th aspect category
α_j^1 α_j^2 α_j^3 α_j^n
Word Representation
v_1 v_2 v_3 ... v_n
(c)

Figure 2: (a) shows the attention-based aspect category classifier, which generates the weights of the words in a sentence about all predefined aspect categories. (b) and (c) show how to generate CAV and CAM based on the weights and the original representations of the words respectively.

3 Method

In this section, we describe our proposed two contextualized aspect category representations, Contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM), in detail.

Motivated by the process that people search for information through search engines: before finding the result they want, they usually try different words and adjust their queries based on previous results, the process to generate CAV or CAM consists of two steps. In the first step, the ACD task as an auxiliary task is used to find coarse aspect category-related information. In the second step, the coarse aspect category-related information is used to optimize original query (e.g. aspect embedding). Specifically, an attention-based aspect category classifier generates the weights of the words in a sentence about all predefined categories. Then the weights are used to generate CAV and CAM. The framework of our proposed method is demonstrated in Figure 2.

3.1 Coarse Aspect Category-related Information

In this step, the ACD task is used to find coarse aspect category-related information. It is a multi-label classification problem, and can be formulated as follows. There are N predefined aspect categories $A = \{A_1, A_2, ..., A_N\}$ in the dataset. Given a sentence, denoted by $S = \{w_1, w_2, ..., w_n\}$, the task checks each aspect $A_j \in A$ to see whether the sentence S mentions it.

An attention-based aspect category classifier is used for this task, because it can offer the weights of the words in a sentence about all predefined categories indicating which word is related to which aspect category. The overall architecture of the model is illustrated in Figure 2 (a). The model contains four modules: embedding layer, LSTM layer, attention layer, and aspect category prediction layer. All aspect categories share the embedding layer and the LSTM layer, and different aspect categories have different attention layers and prediction layers.

Embedding Layer: The input of this layer is a sentence consisting of n words $\{w_1, w_2, ..., w_n\}$. With an embedding matrix U, the input sentence is converted to a sequence of vectors $X = \{x_1, x_2, ..., x_n\}$, where $U \in R^{d\times|V|}$, d is the dimension of the word embeddings, and $|V|$ is the vocabulary size.

LSTM Layer: The word embeddings of the sentence are then fed into a LSTM (Hochreiter and Schmidhuber, 1997) layer, which outputs hidden states $H = \{h_1, h_2, ..., h_n\}$. At each time step i, the hidden state h_i is computed by:

$$h_i = LSTM(h_{i-1}, x_i) \tag{1}$$

The size of the hidden state is also set to be d.

Attention Layer: This layer takes the output of the LSTM layer as input, and produce an attention (Yang et al., 2016) weight vector for each predefined aspect category. Formally, for the j-th aspect category:

$$M_j = tanh(W_j H + b_j) \tag{2}$$

$$\alpha_j = softmax(u_j^T M_j) \tag{3}$$

where $W_j \in R^{d \times d}$,$b_j \in R^d$,$u_j \in R^d$ are learnable parameters, and $\alpha_j \in R^n$ is the attention weight vector. **We can see u_j as aspect embedding, which is the initial query for aspect category-related information.**

Aspect Category Prediction Layer: We use the weighted hidden state as the sentence representation for ACD prediction. For the j-th category:

$$r_j = H\alpha_j^T \tag{4}$$

$$\hat{y}_j = sigmoid(W_j r_j + b_j) \tag{5}$$

where $W_j \in R^{d \times 1}$ and $b_j \in R$.

Loss: As each prediction is a binary classification problem, the loss function for the N aspect categories of the sentence is defined by:

$$L(\theta) = -\sum_{j=1}^{N} y_j log\hat{y}_j + (1 - y_j)log(1 - y_j) + \lambda||\theta||_2^2 \tag{6}$$

where y_j is the correct label, λ is the L_2 regularization factor, N is the number of total aspect categories and θ contains all the parameters.

3.2 Context-dependent Aspect Category Representations

In this step, the attention weight vectors offered by the ACD task is used to generate contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM). **They are the results of optimizing the initial query based on context-dependent information**. Figure 2 (b) and Figure 2 (c) show how to generate CAV and CAM respectively. Given a sentence representation $V = \{v_1, v_2, \ldots, v_n\}$ from an ACSC model and the attention weight vectors of all predefined aspect categories offered by the ACD task, CAV of the j-th aspect category is computed by:

$$v_{CAV_j} = [v_{CAVG_j}; v_{CAVL_j}] \tag{7}$$

$$v_{CAVL_j} = \sum_{i=1}^{n} v_i \alpha_j^i \tag{8}$$

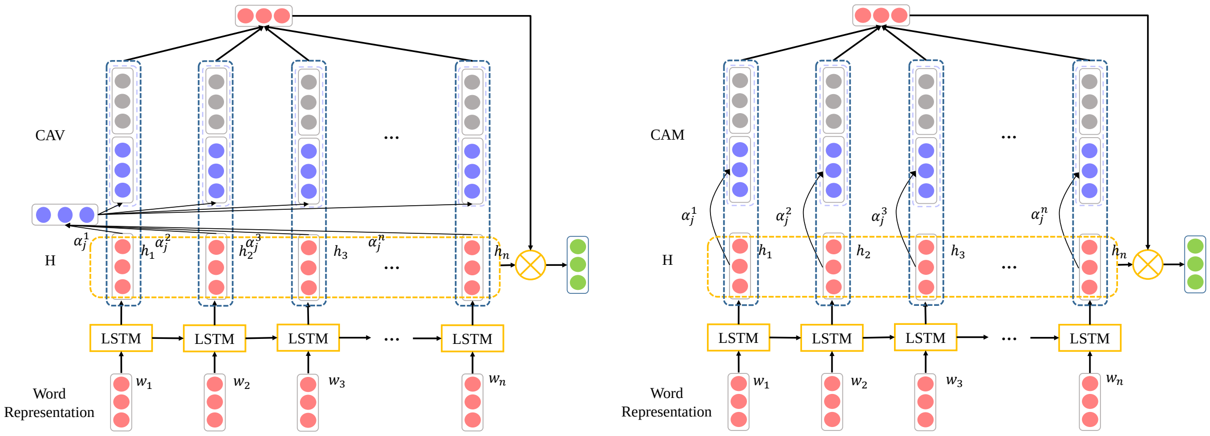
where $v_i \in R^{d_l}$ and d_l is the dimension of the word representations, $v_{CAVG_j} \in R^{d_g}$ and $v_{CAVL_j} \in R^{d_l}$ are the global representation and the local representation respectively, d_g is the dimension of the global aspect category representation, v_{CAVG_j} is initialized randomly and learned during training ACSC models like aspect embedding, and α_j^i indicates the weight of the i-th word about the j-th aspect category. V can be the output of the embedding layer or the sentence encoder in ACSC models.

Because the aspect category representation vectors, such as aspect embedding, often are repeated as many times as there are words in the sentence and concatenated to the word representations of the sentence, we also propose the Contextualized Aspect Matrix (CAM), which can be directly concatenated to the word representations and retains more details of the words. For the j-th aspect category, M_{CAM_j} is computed by:

$$M_{CAM_j} = \{[v_{CAVG_j}; v_1\alpha_j^1], [v_{CAVG_j}; v_2\alpha_j^2], ..., [v_{CAVG_j}; v_n\alpha_j^n]\} \tag{9}$$

where v_{CAVG_j} is the same as it in CAV.

Then the CAV or CAM as queries are used to search for fine-grained aspect category-related information like aspect embedding by ACSC models. Figure 3 shows how to integrate CAV and CAM into AT-LSTM (Wang et al., 2016).



(a) AT-LSTM-CAV

(b) AT-LSTM-CAM

Figure 3: AT-LSTM-CAV and AT-LSTM-CAM, which are obtained by replacing the aspect embedding in AT-LSTM (Wang et al., 2016) with CAV and CAM respectively.

4 Experiments

4.1 Datasets

In order to evaluate the effectiveness of our methods, we conduct experiments on the SemEval-2014 Restaurant Review (Restaurant-2014) dataset (Pontiki et al., 2014) and the Multi-Aspect Multi-Sentiment for Aspect Category Sentiment Analysis (MAMS-ACSA) dataset (Jiang et al., 2019). The Restaurant-2014 is a widely used dataset. However, most sentences in Restaurant-2014 contain only one aspect category or multiple aspect categories with the same sentiment polarity, which makes ABSA task degenerate to sentence-level sentiment analysis. To mitigate the problem, Jiang et al. (2016) released the MAMS-ACSA dataset, all sentences in which contain multiple aspects with different sentiment polarities. Since there is no official development set for the Restaurant-2014 dataset, we use the split offered by Xue et al. (2018). Statistics of these two datasets are given in Table 1.

Dataset		Positive	Negative	Neutral	Total
Restaurant-2014	Train	1855	733	430	3018
	Validation	324	106	70	500
	Test	657	222	94	973
MAMS-ACSA	Train	1929	2084	3077	7090
	Validation	241	259	388	888
	Test	245	263	393	901

Table 1: Statistics of the datasets.

4.2 Implementation Details

We implement our models in PyTorch (Paszke et al., 2017). For all models, including the aspect category classifier and the aspect-category sentiment classification models, we use the pre-trained 300d Glove embeddings (Pennington et al., 2014) to initialize word embeddings, which is fixed in all models. We use Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001 to train all models. We set L_2 regularization factor $\lambda = 0.00001$. The batch sizes are set to 32 and 64 for the Restaurant-2014 dataset and the MAMS-ACSA dataset respectively. For CAV and CAM, d_g is equivalent to d_l. For the aspect category sentiment classification models, we replace the aspect embedding with the CAV or CAM, just adjust the parameters to make the dimensions matching, and use hyper-parameter settings described in original papers. The aspect category classifier and the aspect-category sentiment classification models are trained in a pipeline manner. That is to say, the aspect category classifier is first trained, then the aspect-category sentiment classification models are trained, where the attention weights offered by the aspect category classifier are used to generate CAV or CAM. We fine-tune the hyper-parameters for all baselines on the validation set. We run all models for 5 times and report the average results on the test datasets.

4.3 Comparison Methods

We select the following methods as baseline models:

AE-LSTM (Wang et al., 2016) first get the aspect-aware sentence embedding by concatenating the aspect embedding with each word embedding. Then the aspect-aware sentence embedding is fed into a LSTM layer. The final sentence representation is the last hidden state of the LSTM layer.

AT-LSTM (Wang et al., 2016) models the sentence via a LSTM model. Then it combines the hidden states from the LSTM with the aspect embedding to generate the attention vector. The final sentence representation is the weighted sum of the hidden states.

ATAE-LSTM (Wang et al., 2016) further extends AT-LSTM by taking the aspect-aware sentence embedding as input.

CapsNet (Jiang et al., 2019) is a capsule network that can model the complicated relationship between aspect categories and contexts and obtains state-of-the-art performance on the MAMS-ACSA dataset. It also takes the aspect-aware sentence embedding as input.

Our methods:

***-CAV** replace the aspect embedding in the baseline models with CAV.

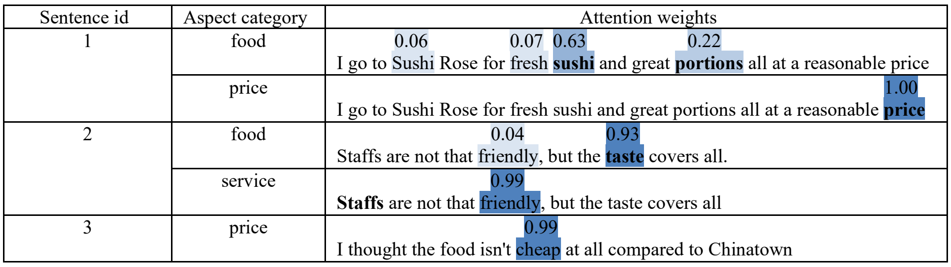
***-CAM** replace the aspect embedding in the baseline models with CAM

Method	Restaurant-2014	MAMS-ACSA
AE-LSTM	76.876 (±2.037)	63.019 (±2.318)
AE-LSTM-CAV	76.711 (±0.963)	66.970 (±0.824)
AE-LSTM-CAM	80.493 (±1.422)	70.721 (±0.717)
AT-LSTM	77.9∗	66.436†
AT-LSTM-CAV	**81.891 (±0.493)**	73.052 (±1.551)
AT-LSTM-CAM	80.740 (±0.681)	**75.539 (±0.657)**
ATAE-LSTM	77.8∗	70.634†
ATAE-LSTM-CAV	81.172 (±0.398)	73.141 (±1.499)
ATAE-LSTM-CAM	81.829 (±0.784)	73.452 (±1.217)
CapsNet	81.110 (±0.492)	73.986†
CapsNet-CAV	77.246 (±0.696)	69.700 (±0.659)
CapsNeT-CAM	80.417 (±0.558)	75.117 (±0.203)

Table 2: Results of the ACSC task in terms of accuracy (%). "∗" refers to citing from Tay et al. (2018). "†" refers to citing from Jiang et al.(2019). Best scores are marked in bold.

4.4 Results and Analysis

Experimental results are illustrated in Table 2. From Table 2 we draw the following conclusions. First, we observe that most models with CAV obtain better performance. Specifically, by replacing the aspect embedding with CAV, our proposed methods outperform their counterparts in 5 of 8 results. compared original models, AT-LSTM-CAV and ATAE-LSTM-CAV improves the performance by 3.9% and 3.4% on the Restaurant-2014 dataset respectively. AE-LSTM-CAV, AT-LSTM-CAV and ATAE-LSTM-CAV improves the performance by 3.9%, 6.6% and 2.5% on the MAMS-ACSA dataset respectively. In addition, AT-LSTM-CAV obtains the best performance on Restaurant-2014. Second, most models with CAM also obtain better performance. Specifically, by replacing the aspect embedding with CAM, most of our proposed methods outperform their counterparts. AE-LSTM-CAM, AT-LSTM-CAM and ATAE-LSTM-CAM improves the performance by 3.6%, 2.8% and 4% on the Restaurant-2014 dataset, by 7.7%, 9.1% and 2.8% on the MAMS-ACSA dataset, respectively. AT-LSTM-CAM and CapsNeT-CAM surpass the



Sentence id	Aspect category	Attention weights
1	food	0.06 0.07 0.63 0.22 I go to Sushi Rose for fresh **sushi** and great **portions** all at a reasonable price
	price	1.00 I go to Sushi Rose for fresh sushi and great portions all at a reasonable **price**
2	food	0.04 0.93 Staffs are not that friendly, but the **taste** covers all.
	service	0.99 **Staffs** are not that friendly, but the taste covers all
3	price	0.99 I thought the food isn't cheap at all compared to Chinatown

Figure 4: Visualization of attention weights of different aspect categories in the ACD task. The numbers on the top of words are the attention weights of the words. The weights greater than 0.01 are labeled. The bold words are the labeled aspect terms. The color depth expresses the important degree of the word.

state-of-the-art baseline mode CapsNeT (+1.6% and +1.1% respectively) on the MAMS-ACSA dataset. Third, CAM outperform CAV in 7 of 8 results. This is because CAM retains more details of the words. Finally, we observe that, in 4 of 6 results, CAV leads to performance drop when aspect category sentiment classification models use it to get aspect-aware sentence embedding by concatenating it with each word embedding. Specifically, compared to AE-LSTM, AT-LSTM-CAV and CapsNet, AE-LSTM-CAV, ATAE-LSTM-CAV and CapsNet-CAV reduce by 0.2%, 0.7% and 4.6% on the Rest14 dataset. Compared to CapsNet, CapsNet-CAV reduces by 4.2% on the MAMS-ACSA dataset. The possible reason is that, in this situation, every word representation contains all aspect category-related information of the sentence, which leads to the sentence encoder, such as LSTM (Hochreiter and Schmidhuber, 1997), to concentrate on the aspect category-related information and discard the aspect category-related sentiment information. It suggests that CAV be best used in attention mechanisms.

4.5 Attention Visualizations

Figure 4 displays the performance of the attention to find aspect category-related words for the ACSC task. Sentence 1 shows that the attention can find the aspect terms for different aspect categories obviously. In sentence 2, while the aspect term for the aspect category *service* is “taste”, the attention finds “friendly” that is more useful than “taste” for the ACSC task. The sentence 3 don’t have any aspect term for the aspect category *price*, however, the attention also finds the useful word “cheap”.

5 Conclusion

In this paper, we propose two novel contextualized aspect category representations, Contextualized Aspect Vector (CAV) and Contextualized Aspect Matrix (CAM). They include both the global information and local information about the aspect category and are better queries to search for aspect category-related information for the ACSC task. Moreover, CAV or CAM contain context-dependent information even though there are no aspect terms in sentences, and aspect term annotation information is not required to generate them. We experiment with several representative aspect embedding-based models by replacing the aspect embedding with CAV or CAM. Experimental results on the SemEval-2014 Restaurant dataset and the Multi-Aspect Multi-Sentiment (MAMS) dataset show that the variants with CAV or CAM obtain better performance. In future works, we will explore the performance of CAV and CAM with knowledge from open knowledge graphs on the ACSC task.

References

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 97–106.

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

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. Can: Constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4593–4602.

Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. 2016. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6281–6286.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deep memory networks for attitude identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 671–680.

Bin Liang, Jiachen Du, Ruifeng Xu, Binyang Li, and Hejiao Huang. 2019a. Context-aware embedding for targeted aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4678–4683, Florence, Italy, July. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019b. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5572–5584.

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

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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, Dublin, Ireland, August. Association for Computational Linguistics.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. SentiHood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1114.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385.

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

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In *The World Wide Web Conference*, pages 2033–2044.

Bowen Xing, Lejian Liao, Dandan Song, Jingang Wang, Fuzheng Zhang, Zhongyuan Wang, and Heyan Huang. 2019. Earlier attention? aspect-aware lstm for aspect sentiment analysis. *arXiv preprint arXiv:1905.07719*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.