

# Multi-Entity Aspect-Based Sentiment Analysis with Context, Entity and Aspect Memory

Jun Yang, Runqi Yang, Chongjun Wang, Junyuan Xie

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China  
yangjunny@126.com, runqiyang@gmail.com, chjwang@nju.edu.cn, jyxie@nju.edu.cn

## Abstract

Inspired by recent works in Aspect-Based Sentiment Analysis (ABSA) on product reviews and faced with more complex posts on social media platforms mentioning multiple entities as well as multiple aspects, we define a novel task called Multi-Entity Aspect-Based Sentiment Analysis (ME-ABSA). This task aims at fine-grained sentiment analysis of (entity, aspect) combinations, making the well-studied ABSA task a special case of it. To address the task, we propose an innovative method that models Context memory, Entity memory and Aspect memory, called CEA method. Our experimental results show that our CEA method achieves a significant gain over several baselines, including the state-of-the-art method for the ABSA task, and their enhanced versions, on datasets for ME-ABSA and ABSA tasks. The in-depth analysis illustrates the significant advantage of the CEA method over baseline methods for several hard-to-predict post types. Furthermore, we show that the CEA method is capable of generalizing to new entity-aspect combinations with little loss of accuracy. This observation indicates that data annotation in real applications can be largely simplified.

## Introduction

Mining massive online social media text gives insights on consumer needs as well as their product experience, helping producers to improve their products. Sentiment Analysis is a useful tool to extract consumers' attitudes towards brands, products as well as related aspects.

Online posts associated with products or services can be divided into two kinds. First is **product reviews**, such as comments on Amazon and Yelp. These posts are about some aspects of particular products or services. In some cases, one post is targeted at one entity: the product or service itself. The second kind is **experience-sharing posts**, such as posts on Twitter or in forums like baby care forums. In most cases, a variable number of entities and aspects are mentioned in a post. For example, *"Tried Pampers. No leakage found but his butt went red. Then I changed to Kao. It's a bit expensive but not allergenic."*

In this post, there are two entities: Pampers and Kao (two paper diaper brands) and three aspects: anti-leakage, anti-allergy and price.

Sentiment analysis over product reviews has drawn much attention in recent years. One of the most widely applied technique is called aspect-based sentiment analysis (ABSA for short, also known as aspect level sentiment analysis) (Wang et al. 2016; Lipenkova 2015; Tang, Qin, and Liu 2016). *Given text and target aspects, the goal of ABSA is to find sentiment polarity towards each specific aspect.* For example, given *"The food is nice but too expensive."* and aspects "food" and "price", ABSA will give results (food, positive), (price, negative).

However, ABSA is not directly applicable to experience-sharing posts, in which the number of entities mentioned may vary. So we define the task of **Multi-Entity Aspect-Based Sentiment Analysis (ME-ABSA)** to model this scenario: *Given entities as well as aspects mentioned in the text<sup>1</sup>, the goal is to predict sentiment polarity towards each (entity, aspect) combination.* In the Pampers-Kao example, we will predict sentiment towards 6 (entity, aspect) combinations for there are 2 entities and 3 aspects. It's obvious that ABSA is a special case of ME-ABSA with the number of entities limited to one.

ME-ABSA is a challenging task. There are three main challenges:

**Aspect sentiment matching:** to extract aspect level sentiment is the basic challenge confronted with both ABSA and ME-ABSA task. In the Pampers-Kao example, "a bit expensive" refers to (price, negative) and "no more allergic" refers to (anti-allergy, positive).

**Entity aspect-sentiment matching:** matching the right aspect and sentiment to the right entity is challenging, especially when there are multiple entities, multiple aspects and multiple sentiment polarities in a post. In the Pampers-Kao example, sentiment towards the same aspect anti-allergy is negative for Pampers but positive for Kao.

**Not-to-match:** this mainly associates with neutral posts. In the Pampers-Kao example, we cannot match (anti-leakage, positive) to Kao. Kao should remain neutral on anti-leakage aspect.

In this paper, we introduce an innovative method for this task. We use an interaction layer, a position attention layer

<sup>1</sup>Extracting entities and aspects mentioned in the post lies in the scope of information extraction, which is not the key concern in this sentiment analysis task.

and an LSTM layer with attention to build context memory. The interaction layer accepts word vectors, the entity vector and the aspect vector as input, and performs element-wise multiplication and concatenation to fuse entity, aspect and context information. The position attention layer is designed to utilize position information of the entity and the aspect. The LSTM layer with attention transforms the input to context memory as a vector. We then iteratively update the entity memory and the aspect memory with the context memory, so we get sentimental as well as contextual entity and aspect representations after this step. Finally, we use a linear layer followed by a softmax layer to predict sentiment polarity to the given (entity, aspect) combination.

Experiments show that our method outperforms several baselines, including the state-of-the-art of ABSA task, and their enhanced versions on the dataset for ME-ABSA task. To validate the effectiveness of our method, we also make comparisons on the special case of ME-ABSA: the ABSA task. Our method outperforms the state-of-the-art method on all three ABSA benchmark datasets. In the in-depth analysis, we validate the effectiveness of each component and further divide the data into seven types and show the significant advantage of our method in several hard-to-predict types. We also carry out experiments on new and rare (entity, aspect) combinations, indicating the capability of the method and the potential to lessen the burden of data annotation in real applications.

The contributions of this paper include:

- We define **Multi-Entity Aspect-Based Sentiment Analysis (ME-ABSA)** task, which is suited for fine-grained sentiment analysis towards (entity, aspect) combinations, making Aspect-Based Sentiment Analysis (ABSA) a special case of ME-ABSA, with the number of entities limited to one.
- We propose to model context, entity and aspect memory for ME-ABSA task. Experiments show that our method CEA outperforms baseline methods in ME-ABSA dataset, ABSA dataset and hard-to-predict post types in ME-ABSA dataset.
- We release our dataset, looking forward to advancing the research in fine-grained sentiment analysis.

## Related Work

Aspect-based sentiment analysis, ABSA for short, is a subdomain in sentiment analysis (Liu 2012) which focuses on fine-grained sentiment information. There are two main approaches to solve ABSA problem. The first one is the traditional method using lexicons and rules. (Hu and Liu 2004) summed up sentiment scores of all sentiment words. (Ding, Liu, and Yu 2008) proposed a holistic lexicon-based approach considering both explicit and implicit opinions. This method is further improved by identifying the aspect-opinion relations using tree kernel method (Nguyen and Shirai 2015a). The second one is the machine learning method. (Ramesh et al. 2015) employed hinge-loss Markov random fields to tackle aspect sentiment problem in MOOC. (Wang and Ester 2014) proposed a sentiment-aligned topic model for product aspect rating prediction. (Kiritchenko et

al. 2014) combined lexicon-based method and feature-based SVM, and took the first place in detecting sentiment towards aspect categories in SemEval 14 competition. (Nguyen and Shirai 2015b) constructed target dependent binary phrase dependency tree to build the representation of an aspect. (Tang et al. 2016) solved the problem with recurrent neural network, and proposed two methods TD-LSTM and TC-LSTM. (Wang et al. 2016) proposed an attention-based LSTM method. It is the best method among the ones who only deals with abstractive context memory. (Tang, Qin, and Liu 2016) introduced a deep memory network based method to solve ABSA task, and achieved state-of-the-art performance.

Entity sentiment detection is another subdomain in sentiment analysis. (Moilanen and Pulman 2009) proposed a framework for modeling entity-level sentiment. It suggests that compositional sentiment parsing can generate effective results. (Deng and Wiebe 2015) performed entity/event-level sentiment analysis using probabilistic soft logic models. (Mitchell et al. 2013) modeled entity sentiment detection as a sequence labeling problem. (Li and Lu 2017) introduced the notion of sentiment scopes and proposed methods capable of learning sentiment scopes to jointly predict named entities and their associated sentiment.

When performing entity or aspect sentiment classification, entity or aspect is assumed given. We keep the same assumption. There are many solutions for aspect extraction, such as (Mukherjee and Liu 2012), (Liu 2014), (Poria et al. 2014), and many solutions for entity extraction, such as (Etzioni et al. 2005), (Gupta and Manning 2014).

## Baby Care Dataset

In this section, we introduce our dataset for ME-ABSA task. Our dataset is mainly about *baby care(BC)*, including topics in diaper, milk powder, disease and so on. We collected the posts on [www.babytree.com](http://www.babytree.com), one of the biggest baby care forums in China. Entities in our experiment include category names (like milk powder, diaper), brand names and product names. Entities and aspects are listed by professionals in maternal and child industry.

Three native speakers were invited to annotate the sentiment labels given the post and all combinations of (entity, aspect) mentioned in the post. Two weeks, including time for the cross-check, are given to these native speakers to annotate the data. We provide a training set BC-Train for model training, a development set BC-Dev for parameter tuning and a test set BC-Test for evaluation. The statistics of our dataset are listed in Table 1.

The entity and aspect given in the dataset are entity category and aspect category, not entity term or aspect term mentioned in the post. For example, in the Pampers-Kao example, aspect anti-allergy is mentioned as terms “butt went red” and “not allergenic”. In the dataset, we provide entity positions and some of the aspect positions. For all the data, entity position is given, and for around one-fourth of the data, aspect position is not given for the difficulty of annotating the exact position of the aspect.

For fine-grained evaluation, we also provide seven types of labels to each post. For example,  $E^1A^+S^1$

Information	BC-Train	BC-Dev	BC-Test
data count	29354	3682	3677
# entities	115	111	110
# aspects	166	122	119
# polarities of sentiment	3	3	3
positive%, neutral%, negative%	51%, 37%, 12%	50%, 39%, 11%	50%, 38%, 12%
# only entity position	6805	868	860
# both position	22549	2814	2817
post length(min)	4	5	4
post length(max)	283	280	284
post length(avg)	66	68	64
#(%) $E^1 A^1 S^1$	1298(4%)	157(4%)	179(5%)
#(%) $E^1 A^+ S^1$	4485(15%)	497(14%)	564(15%)
#(%) $E^1 A^+ S^+$	2152(7%)	307(8%)	293(8%)
#(%) $E^+ A^1 S^1$	844(3%)	125(3%)	91(3%)
#(%) $E^+ A^1 S^+$	1197(4%)	167(5%)	152(4%)
#(%) $E^+ A^+ S^1$	2243(8%)	328(9%)	154(4%)
#(%) $E^+ A^+ S^+$	17135(59%)	2101(57%)	2244(61%)

Table 1: Statistics on the Baby Care dataset. The statistics is on the basis of (entity, aspect, sentiment) triples. We count each triple as a piece of data, same as the common practice in ABSA task.

refers to single-entity, multi-aspect and single-sentiment-polarity posts. Upper case “E” stands for entity, “A” stands for aspect and “S” stands for sentiment. Super-script “1” stands for single and “+” represents multiple. Following the same rule, we also define  $E^1 A^1 S^1$ ,  $E^1 A^+ S^+$ ,  $E^+ A^1 S^1$ ,  $E^+ A^1 S^+$ ,  $E^+ A^+ S^1$  and  $E^+ A^+ S^+$ . Note that we don’t have single-entity, single-aspect and multi-sentiment-polarity ( $E^1 A^1 S^+$ ) posts in our dataset, which meets our common sense.

It is noteworthy that the number of positive samples is more than negative samples as users prefer to recommend products rather than criticize them in this forum. The percentage of neutral samples in our dataset is more than one third. The high percentage of neutral samples is different from some sentiment analysis datasets. It is mainly because in this task, if there is no attitude expressed towards the (entity, aspect) combination mentioned in the text, the sentiment is neutral.

## The CEA Method

### An Overview of the Architecture

Our method models Context, Entity and Aspect memory, called CEA method. The memory here refers to the representation that is abstractive, recallable and modifiable. The CEA method consists of 3 modules: context memory encoding, entity and aspect memory updating and sentiment prediction. Figure 1 gives an illustration of the architecture.

### Context Memory Encoding

The goal of encoding the context memory is to get a good representation of the context given the entity and aspect information. This module accepts word vector, entity vector and aspect vector as input, models interaction between the word vector, the entity vector and the aspect vector respectively, selectively focuses on the important information given position information, feeds the output to LSTM layer and calculates context memory with attention mechanism.

**Module input.** The input of the Context Encoder module contains three parts: pre-trained word vectors of each word  $w_i$  in the post, the entity vector  $v_e$  and the aspect vector  $v_a$ . If entity(aspect) is a single word,  $v_e(v_a)$  is initialized by the word vector of the entity(aspect), otherwise, it is calculated by averaging word vectors of each word in the entity(aspect). For example, aspect vector for “brain development” is the average of the word vector of “brain” and the word vector of “development”.

**Interaction layer A.** The interaction layer aims to take entity and aspect information into consideration when encoding context memory. There are three choices:

1. no interaction:

$$f(w_i, v_e, v_a) = w_i \quad (1)$$

2. concatenation:

$$f(w_i, v_e, v_a) = [w_i; v_e; v_a] \quad (2)$$

3. element-wise multiplication and concatenation:

$$f(w_i, v_e, v_a) = [w_i; w_i \odot v_e, w_i \odot v_a] \quad (3)$$

Here  $\odot$  is element-wise multiplication. (Tang et al. 2016) used no interaction in the paper. (Wang et al. 2016) argues that concatenation is better than no interaction in ABSA task. Our experiments showing that element-wise multiplication and concatenation is the best among them.

**Position attention layer.** Based on the idea that sentiment towards the entity and aspect is more likely to be expressed by the words near them. The model pays attention to each word and the attention rate is based on the position attention function  $g_e$  for entity and  $g_a$  for aspect. Two facts make the functions more complex: an entity or aspect may be mentioned various times in the same post; position may be unknown.

$$g_e(p_w^i, p_e^i, N) = \begin{cases} 1 - (|p_w^i - p_e^i|) / N & p_e \text{ known,} \\ 1 & p_e \text{ unknown.} \end{cases} \quad (4)$$

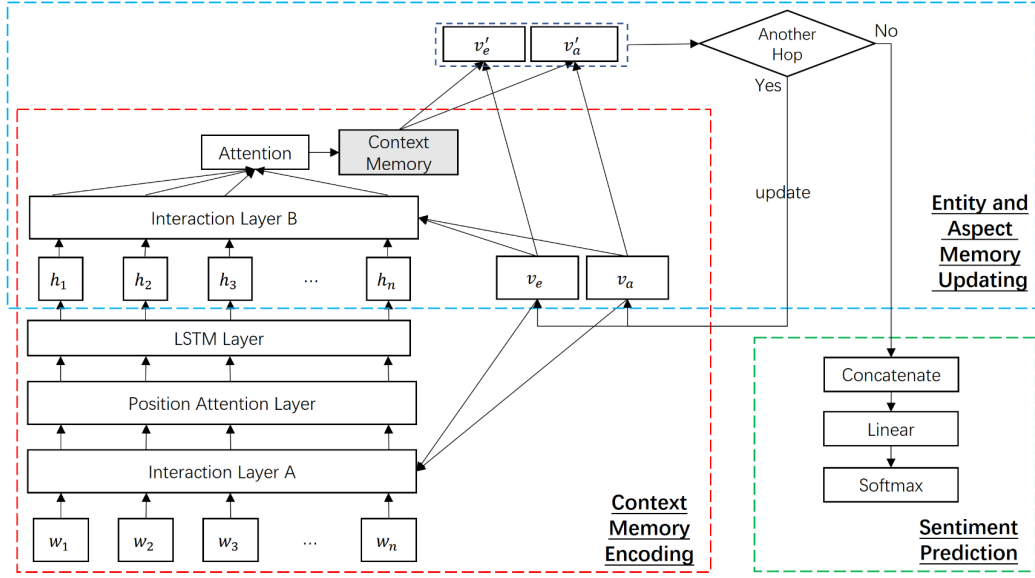


Figure 1: Architecture of CEA.

$$g_a(p_w^i, p_a^i, N) = \begin{cases} 1 - (|p_w^i - p_a^i|) / N & p_a \text{ known,} \\ 1 & p_a \text{ unknown.} \end{cases} \quad (5)$$

$p_w^i$  is the position of  $i^{th}$  word,  $p_e$  is the position of entity term,  $p_a$  is the position of aspect term.  $p_e^i$  is the position of the entity term nearest to  $i^{th}$  word,  $p_a^i$  is the position of the aspect term nearest to  $i^{th}$  word,  $N$  is the length of the post.

Then, the output after position attention layer is:

$$\begin{aligned} out_i &= f(w_i, v_e, v_a) * g_e(p_w^i, p_e^i, N) \\ &\quad * g_a(p_w^i, p_a^i, N), i \in [1, N] \end{aligned} \quad (6)$$

We also tried other forms of position attention, this one performs the best.

**LSTM layer.** We feed the output of position attention layer to Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber 1997). LSTM is a powerful variant of recurrent neural networks which is able to process sequences of arbitrary length and capture long-range dependencies through gated mechanism. LSTM outputs hidden states  $\{h_i\}$  for  $i \in [1, N]$ . Note that, we perform dropout before LSTM. Dropout is useful to prevent overfitting. We set dimension of hidden states to the dimension of word vector  $d$  for future use.

**Interaction layer B.** Same as interaction layer A except for changing  $w_i$  to  $h_i$ . The output of interaction layer B is  $[h_i; h_i \odot v_e; h_i \odot v_a]$

**Attention.** We use attention mechanism to calculate the context memory. Attention is calculated as:

$$att_i = W_1 \tanh(W_2 [h_i; h_i \odot v_e; h_i \odot v_a]) + b_1, \quad i \in [1, N] \quad (7)$$

$$\alpha = \text{softmax}(att) \quad (8)$$

$W_1, W_2$  and  $b_1$  are parameters.  $\tanh$  is the activation function. At last, the context memory is calculated as:

$$c = \sum_{n=1}^N \alpha_n h_i \quad (9)$$

### Entity and Aspect Memory Updating

The goal of updating entity and aspect memory is to transform the original entity and aspect vector, initialized by word vectors, to the entity and aspect memory containing both sentimental and contextual information.

**Update entity and aspect memory.** In this step, we update entity memory and aspect memory separately. We provide two updating methods:

1. simple summation:

$$\begin{aligned} v'_e &= c + v_e \\ v'_a &= c + v_a \end{aligned} \quad (10)$$

2. weighted summation:

$$\begin{aligned} v'_e &= w_c * c + w_e * v_e \\ v'_a &= w_c * c + w_a * v_a \end{aligned} \quad (11)$$

Weighted summation is more flexible than simple summation, but increases model complexity at the same time. We experimented with both updating methods, simple summation performs better in accuracy.

**Multi-Hop for Updating.** It is widely accepted that models with multiple layers are able to learn multiple levels of abstraction (LeCun, Bengio, and Hinton 2015). (Tang, Qin, and Liu 2016) showed deep memory networks with 7-9 layers performs better on ABSA task. In this work, we use multi-hops to learn abstractive representation for both entity and aspect. A new hop contains steps including: update

entity and aspect vector by  $v_e = v'_e$  and  $v_a = v'_a$ , calculate interaction layer B, update context memory with attention, update entity and aspect memory and judge if stop condition is satisfied. We just fix the number of hops in our experiments. We tune maximum number of hops in the parameter tuning procedure.

## Sentiment Prediction

The updated entity and aspect memory is regarded as the representation of “the sentiment towards the entity and aspect in the context”. We concatenate the updated entity and aspect memory vector and feed it to a linear layer followed by a softmax layer to predict the sentiment polarity as the final output. Note that, we perform another dropout before softmax.

The loss function of our method is defined as the summation of cross entropy loss and  $l_2$  regularization loss.

## Experiment

We describe our experimental methodology, quantitative and qualitative results which validate the effectiveness of our method for both ME-ABSA task and its special case: ABSA task.

### Experiment Setup

Word vectors are pre-trained with Glove (Pennington, Socher, and Manning 2014) and dimension is set to 300. We used Jieba<sup>2</sup> for Chinese word segmentation. LSTM hidden state size is set to 300. After parameter tuning on BC-Dev dataset, final parameter settings of our method CEA are as follows. We train our model 10 iterations with a batch of 25 instances, the  $L_2$ -regularization weight of 0.001, the learning rate of 0.001 for Adam optimizer. The dropout rate before LSTM and before softmax are both set to 0.5. The maximum hop count is set to 3. To overcome out-of-vocabulary(OOV) problem, we initialize OOV words with random word vectors with random uniform ranging from -0.01 to 0.01.<sup>3</sup>

Our evaluation metrics include: **accuracy**, **precision**, **recall**, **f1-score**: reflecting the effectiveness, and **testing time**: evaluating the efficiency. Testing time is the running time on test set BC-Test on an i7-16GB-GTX1070(GPU) computer with tensorflow framework<sup>4</sup> and batch size at testing is set to 1000 for all methods. We only calculate running time of the method, running time of some common parts such as pre-processing, training, writing results to file are not included.

### Comparison Methods

We compare our method CEA with the following baseline methods.

(1) **LSTM**: Long short-term memory networks for ME-ABSA. LSTM is often regarded as one of the baselines in NLP tasks. We use word vector as input to LSTM layer,

concatenate entity aspect vectors with the last LSTM hidden state, and use a linear layer and a softmax layer to predict sentiment polarity.

(2) **ATAE-LSTM**: the method proposed by (Wang et al. 2016). It is the best one in ABSA task among methods only updates context memory including LSTM(Tang et al. 2016), TD-LSTM(Tang et al. 2016), TC-LSTM(Tang et al. 2016), AE-LSTM(Wang et al. 2016).

(3) **DMN**: the state-of-the-art method on ABSA task proposed by (Tang, Qin, and Liu 2016). This work only updates abstractive aspect memory without updating context memory.

(4) **ATAE-LSTM++**: an enhanced version of ATAE-LSTM. As ATAE-LSTM proposed in (Wang et al. 2016) only deals with aspect part, we add entity part in the same manner as aspect. The main difference from our method is ATAE-LSTM++ only updates context memory for prediction.

(5) **DMN++**: an enhanced version of DMN. As DMN proposed in (Tang, Qin, and Liu 2016) only deals with aspect part, we add entity part in the same manner as aspect. The main difference from our method is DMN++ only updates entity and aspect memory for prediction.

(6) **OracleE**: instead of comparing our method with entity sentiment classification methods such as(Moilanen and Pulman 2009; Li and Lu 2017), we assume there is a method named OracleE who has 100% accuracy in predicting entity sentiment. However, OracleE may get confused when there are multiple aspects related to a single entity and the sentiment polarities towards each aspect are different. For post types including “ $E^1 A^1 S^1$ ,  $E^1 A^+ S^1$ ,  $E^+ A^1 S^1$ ,  $E^+ A^1 S^+$  and  $E^+ A^+ S^1$ ”, we suppose OracleE can predict them all right. For “ $E^1 A^+ S^+$  and  $E^+ A^+ S^+$ ” post types, OracleE predicts the same sentiment label to each aspect of the same entity. The predicted sentiment label is the one that maximum occurred on the entity in this post. For example, assume there is an entity in a post having 5 aspects, 2 positive, 2 neutral, 1 negative, we randomly select positive or neutral as the sentiment towards all aspects of this entity. We do not report precision, recall or f1-score for OracleE method for there is random selection. It is obvious that all entity sentiment classification methods perform no better than OracleE.

For baseline methods, we also carried out parameter tuning on the BC-Dev dataset. Most experiment settings are same among methods, including word vectors, dropout, iteration num, batch size,  $L_2$ -regularization weight, learning rate and optimizer. For LSTM, ATAE-LSTM and ATAE-LSTM++, we apply fine-tune for word vectors which improve the accuracy. For DMN, DMN++ and CEA, fine-tune is turned off for they perform better without fine-tune.

## Results and Analysis

Table 2 shows the comparison results of the methods on ME-ABSA task. ATAE-LSTM++ performs better than LSTM, indicating that entity and aspect information for LSTM input as well as attention mechanism are useful for prediction. DMN++ is faster but weaker than ATAE-LSTM++. We think this is because *entity aspect-sentiment matching challenge* mentioned in the introduction section. DMN++, mod-

<sup>2</sup><https://github.com/fxsjy/jieba/>

<sup>3</sup>Code and data are available at <http://www.marcpoint.com/junyang.html>.

<sup>4</sup>[www.tensorflow.org](http://www.tensorflow.org).

Metrics	LSTM	ATAE-LSTM++	DMN++	ATAE-LSTM	DMN	CEA	OracleE
test accuracy	69.62	71.55	68.15	66.77	61.82	<b>80.74</b>	67.47
negative precision	44.00	47.48	52.69	43.45	39.06	<b>70.05</b>	-
negative recall	25.88	31.06	11.53	14.82	5.88	<b>64.94</b>	-
negative f1-score	32.59	37.55	18.92	22.11	10.22	<b>67.40</b>	-
neutral precision	63.44	66.93	62.41	60.90	54.98	<b>76.61</b>	-
neutral recall	73.31	72.02	67.65	70.73	62.20	<b>80.85</b>	-
neutral f1-score	68.02	69.38	64.92	65.45	58.36	<b>78.67</b>	-
positive precision	78.63	78.73	73.03	73.50	67.58	<b>86.42</b>	-
positive recall	76.86	80.46	81.49	75.67	74.06	<b>84.28</b>	-
positive f1-score	77.74	79.58	77.03	74.57	70.67	<b>85.34</b>	-
testing time(s)	0.94	2.03	1.63	1.50	<b>0.89</b>	3.08	-

Table 2: Comparison results on the *baby care(BC)* dataset.

Datasets	ATAE-LSTM	DMN	CEA
restaurant.term	77.79	80.32	<b>80.98</b>
restaurant.catgory	83.42	82.82	<b>84.44</b>
laptop.term	67.18	71.34	<b>72.88</b>

Table 3: Accuracy(%) on ABSA datasets.

ified from DMN for aspect-based sentiment analysis, updates entity and aspect representation mainly by attention, ignoring the word order in the model. This works well in ABSA task(DMN is the state-of-the-art method for ABSA task). When it comes to ME-ABSA task, ignoring the word order leads to the difficulty in context understanding, suffering from *entity aspect-sentiment matching challenge*. ATAE-LSTM and DMN achieve low accuracy, showing that considering entity information is important in ME-ABSA task. The accuracy of OracleE is low, showing aspect information is important and all entity sentiment classification methods will not outperform our method in ME-ABSA task. Our method CEA is significantly better than baseline methods in accuracy and PRF. CEA is more complex, the test time is around 50% more than the second best method ATAE-LSTM++, costing around 3 seconds.

Based on the fact that a) LSTM, ATAE-LSTM++, DMN++ performs better than other baseline methods; b) ATAE-LSTM++ is the enhanced version of LSTM who only updates context memory; c) DMN++ only undates entity and aspect memory, we only compare our method with ATAE-LSTM++ and DMN++ in the following experiments on test set BC-Test.

### Performance on Special Case: ABSA Task

To further validate the capability of our method, we test it on the special case of ME-ABSA: the ABSA task, with benchmark datasets. We experiment on the datasets of SemEval 2014 Task 4<sup>5</sup>. The datasets are customers reviews, one from restaurant domain while the other from laptop domain. The datasets provide one training set and one test set for each domain.

In dataset from restaurant domain, aspect-based sentiment analysis task is further divided into aspect term sentiment

analysis and aspect level sentiment analysis. The former provides context, aspect term mentioned in the context, the position of aspect term and sentiment towards the aspect term. The latter provides context, aspect category and sentiment towards the aspect category. Aspect categories are *food, price, service, ambience, anecdotes/miscellaneous*. For aspect category sentiment classification, aspect position is not given, we remove position attention layer in the experiments. In dataset from laptop domain, there is only aspect term sentiment analysis task.

In these datasets, there are four categories of sentiment label: *positive, neutral, negative* and *conflict*. Same as previous work (Tang, Qin, and Liu 2016), we remove *conflict* category as the number of instances is very tiny.

The word embeddings are 300-dimensional Glove vectors (Pennington, Socher, and Manning 2014) pre-trained on web data with a vocabulary size of 1.9M<sup>6</sup>. We train our model with a batch of 25 instances, the  $L_2$ -regularization weight of 0.001, the learning rate of 0.001 for Adam optimizer. The dropout rate is set to 0.5. We fine-tune word vectors for ATAE-LSTM. DMN and our method run without fine-tune.

Table 3 illustrates the result.<sup>7</sup> Our method outperforms DMN, the state-of-the-art method in ABSA task, showing the effectiveness in modeling context memory. Our method also outperforms ATAE-LSTM, the best ABSA method among those who only model context memory while ignoring aspect memory updating.

The superior results in both ME-ABSA and ABSA datasets validate the effectiveness of modeling the context, entity and aspect memory.

### In-depth Analysis

#### Effectiveness of Components

Table 4 illustrates the accuracy of modifying a component of our method. It validates the effectiveness of some components that modifying any of them may harm the accuracy.

<sup>6</sup>Pre-trained word vectors can be downloaded from <https://nlp.stanford.edu/projects/glove/>

<sup>7</sup>There are small differences from the figure reported in the original paper. We think they are caused by unstated or different settings on: the way of dealing with “conflict” samples, the word vectors used, fine-tune and dropout rate.

<sup>5</sup><http://alt.qcri.org/semeval2014/>

Modified Layer	Modification	Accuracy
interaction A&B	-r concatenation	76.72
position attention	remove layer	80.34
entity/aspect updating	-r weighed summation	79.41
entity/aspect updating	hop count=1	79.08
entity/aspect updating	hop count=2	80.20
entity/aspect updating	hop count=3(CEA)	<b>80.74</b>
entity/aspect updating	hop count=4	79.57
entity/aspect updating	hop count=5	78.44
general	fine tune word vectors	79.85

Table 4: Accuracy(%) on modifying the method. “-r” refers to replace with.

Types	ATAE-LSTM++	DMN++	CEA
$E^1 A^1 S^1$	83.80	72.07	<b>88.27</b>
$E^1 A^+ S^1$	93.79	90.25	<b>95.04</b>
$E^1 A^+ S^+$	66.55	64.85	<b>74.74</b>
$E^+ A^1 S^1$	64.84	59.34	<b>80.22</b>
$E^+ A^1 S^+$	58.55	53.95	<b>76.32</b>
$E^+ A^+ S^1$	83.77	79.87	<b>88.31</b>
$E^+ A^+ S^+$	65.95	63.24	<b>77.14</b>

Table 5: Accuracy(%) on different types of posts.

### Performance on Seven Post Types

We evaluate our method with baseline methods on seven types of data introduced in dataset section to show the capability of methods. Table 5 shows the evaluation results. Generally speaking, all methods achieve relatively lower performance on posts with multiple entities, indicating the challenge in ME-ABSA task.

In all post types, ATAE-LSTM++ performs better than DMN and our method performs the best. For  $E^1 A^+ S^+$ ,  $E^+ A^1 S^1$ ,  $E^+ A^1 S^+$  and  $E^+ A^+ S^+$ , the accuracy of both ATAE-LSTM and DMN are lower than 0.70, so we name them “hard-to-predict post types”. In these four types,  $E^1 A^+ S^+$  is related to aspect sentiment challenge introduced in the introduction section and the other three post types suffer from all three challenges. Our method is significantly better than baseline methods in these hard-to-predict post types.

### Performance on New and Rare Combinations

The motivation of this section is as follows. Assume there are  $m$  entities,  $n$  aspects, for every (entity, aspect) combination, models need  $k$  instances on average to learn well. Can we feed less than  $m * n * k$  pieces of data, to relieve the burden of data annotation in the real applications? As it is hard to give theoretical proof, we try a data-driven method: we validate the accuracy on new and rare (entity, aspect) combinations.

Figure 2 illustrates that there are 93 pieces of data in the test set having (entity, aspect) combinations that do not appear in the training set. They are new combinations. The accuracy of CEA and ATAE-LSTM++ on new combinations is close to the accuracy on the whole dataset. On rare combinations, the accuracy vibrates due to the small amount of rare instances. It can be observed that there is no clear correlation

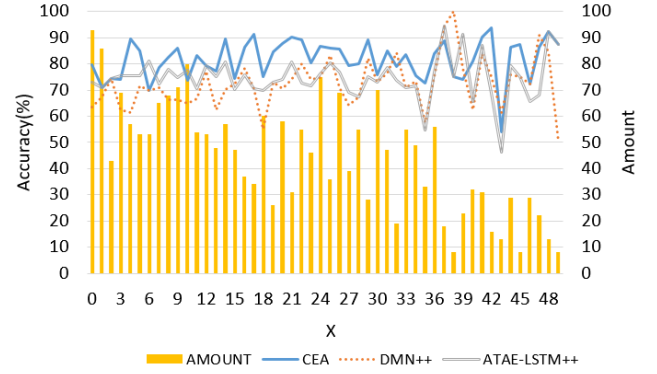


Figure 2: Amount and accuracy of new and rare samples. The horizontal axis X refers to the number of occurrence of an (entity, aspect) combination in the training set. The line chart is related to left vertical axis: accuracy(%). The histogram is related to right vertical axis: the amount of times (entity, aspect) combinations appeared in the test set. For example, there are 93 pieces of data containing new combinations.

between the amount of training data and the performance of our CEA method. This is desirable as the lower the correlation, the better the method is. To further quantify the correlation, we apply Pearson correlation coefficient to measure the correlation between the performance and the amount of training data for the three methods. For our method CEA, the Pearson correlation coefficient is 0.047, ATAE-LSTM++ is -0.077 and DMN++ is 0.319. This indicates DMN++ suffers from lack of training data a bit more than CEA and ATAE-LSTM++. The statistics show a promising result of CEA on new and rare combinations, indicating the unnecessary of annotating all the possible entity aspect combinations.

### Conclusion

In this paper, we define a new task named Multi-Entity Aspect-Based Sentiment Analysis(ME-ABSA) to investigate fine-grained sentiment analysis, making ABSA a special case of this task with the number of entities limited to one. We propose a method CEA with context, entity and aspect memory modeling. Experiments validate the effectiveness of our method. The method performs significantly better than baseline methods on hard-to-predict post types. We also show promising results on new and rare (entity, aspect) combinations, indicating that data annotation in real applications can be largely simplified.

In future work, we will try to modify the stop condition in updating entity and aspect memory. We expect that decreasing hop count for easy samples and increasing hop count for hard ones will improve both the effectiveness and efficiency.

### Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. This paper is supported by the National Key Research and Development Program of China

(Grant No. 2016YFB1001102), the National Natural Science Foundation of China (Grant No.61375069, 61403156, 61502227), the Fundamental Research Funds for the Central Universities (Grant No. 020214380036, 020214380038) and the Collaborative Innovation Center of Novel Software Technology and Industrialization at Nanjing University.

## References

- Deng, L., and Wiebe, J. 2015. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 179–189. Lisbon, Portugal: Association for Computational Linguistics.
- Ding, X.; Liu, B.; and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, 231–240. ACM.
- Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence* 165(1):91–134.
- Gupta, S., and Manning, C. D. 2014. Improved pattern learning for bootstrapped entity extraction. In *CoNLL*, 98–108.
- Hochreiter, S., and Schmidhuber, J. 1997. *Long short-term memory*. Springer Berlin Heidelberg.
- Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. ACM.
- Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 437–442.
- LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436–444.
- Li, H., and Lu, W. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *AAAI*, 3482–3489.
- Lipenkova, J. 2015. A system for fine-grained aspect-based sentiment analysis of chinese. In *ACL (System Demonstrations)*, 55–60.
- Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.
- Liu, Z. C. A. M. B. 2014. Aspect extraction with automated prior knowledge learning. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics*, 347–358.
- Mitchell, M.; Aguilar, J.; Wilson, T.; and Van Durme, B. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1643–1654. Seattle, Washington, USA: Association for Computational Linguistics.
- Moilanen, K., and Pulman, S. 2009. Multi-entity sentiment scoring. In *RANLP*, 258–263.
- Mukherjee, A., and Liu, B. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 339–348. Association for Computational Linguistics.
- Nguyen, T. H., and Shirai, K. 2015a. Aspect-based sentiment analysis using tree kernel based relation extraction. 9042:114–125.
- Nguyen, T. H., and Shirai, K. 2015b. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing*, 2509–2514.
- Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 1532–1543.
- Poria, S.; Cambria, E.; Ku, L.-W.; Gui, C.; and Gelbukh, A. 2014. A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, 28–37.
- Ramesh, A.; Kumar, S. H.; Foulds, J. R.; and Getoor, L. 2015. Weakly supervised models of aspect-sentiment for online course discussion forums. In *ACL (1)*, 74–83.
- Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 3298–3307. Osaka, Japan: The COLING 2016 Organizing Committee.
- Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 214–224. Austin, Texas: Association for Computational Linguistics.
- Wang, H., and Ester, M. 2014. A sentiment-aligned topic model for product aspect rating prediction. In *Conference on Empirical Methods in Natural Language Processing*, 1192–1202.
- Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based lstm for aspect-level sentiment classification. In *Conference on Empirical Methods in Natural Language Processing*, 606–615.