Effective Attention Modeling for Aspect-Level Sentiment Classification

Ruidan He[†‡], **Wee Sun Lee**[†], **Hwee Tou Ng**[†], and **Daniel Dahlmeier**[‡]
[†]Department of Computer Science, National University of Singapore
[‡]SAP Innovation Center Singapore
[†]{ruidanhe,leews,nght}@comp.nus.edu.sg
[‡]d.dahlmeier@sap.com

Abstract

Aspect-level sentiment classification aims to determine the sentiment polarity of a review sentence towards an opinion target. A sentence could contain multiple sentiment-target pairs; thus the main challenge of this task is to separate different opinion contexts for different targets. To this end, *attention mechanism* has played an important role in previous state-of-the-art neural models. The mechanism is able to capture the importance of each context word towards a target by modeling their semantic associations. We build upon this line of research and propose two novel approaches for improving the effectiveness of attention. First, we propose a method for target representation that better captures the semantic meaning of the opinion target. Second, we introduce an attention model that incorporates syntactic information into the attention mechanism. We experiment on attention-based LSTM (Long Short-Term Memory) models using the datasets from SemEval 2014, 2015, and 2016. The experimental results show that the conventional attention-based LSTM can be substantially improved by incorporating the two approaches.

1 Introduction

Aspect-level sentiment classification is an important task in fine-grained sentiment analysis (Pang and Lee, 2008). Given a sentence and an opinion target (also called aspect expression) occurring in the sentence, the task aims to determine the sentiment polarity of the sentence towards the opinion target. An opinion target or target for short refers to a word or a phrase (a sequence of words) describing an aspect of an entity. For example, in the sentence "*This little place has a cute interior decor and affordable prices*", the targets are *interior decor* and *prices*, and they belong to the aspects *ambience* and *price* respectively.

Compared to document-level or sentence-level sentiment classification, the main challenge of aspect-level sentiment classification is to differentiate sentiments towards different targets when there are multiple targets in a sentence. For instance, the sentence "*The appetizers are ok, but the service is slow.*" expresses a neutral sentiment on the target *appetizers* and a negative sentiment on the target *service*. To this end, attention mechanism has played an important role in state-of-the-art neural models for this task. It assigns a positive weight att_i for each context word w_i, which can be interpreted as the probability that w_i is the right word to focus on when inferring the sentiment polarity of the given target. The weight att_i is generally computed as a function of the hidden representation $\mathbf{h_i}$ of w_i and the target representation $\mathbf{t}$ as follows:

$$att_i \propto f_{score}(\mathbf{h_i}, \mathbf{t}) \tag{1}$$

It has been shown that adding an attention model substantially improves the accuracy of aspect-level sentiment classification (Tang et al., 2016b; Wang et al., 2016; Liu and Zhang, 2017).

Our work builds upon this line of research. We propose two novel approaches for improving the effectiveness of attention models. The first approach is a new way of encoding a target which better captures the aspect semantics of the target expression. The target representation is crucial since attention weights are computed based on it as shown in Eq. 1. In representing the target, we are mapping a word or

Proceedings of the 27th International Conference on Computational Linguistics, pages 1121–1131
Santa Fe, New Mexico, USA, August 20-26, 2018.

a phrase into a vector in $\mathbb{R}^d$. Ideally, targets that are semantically similar should be mapped to vectors that are close together in $\mathbb{R}^d$. However, previous neural attention models simply map a target by averaging its component word vectors. This may work fine for targets that only contain one word but may fail to capture the semantics of more complex expressions, as also mentioned by Tang et al. (2016b). For example, we cannot obtain a good representation for "*hot dog*" by averaging the word vectors of "*hot*" and "*dog*". Hot would be close to words like warm or cold and dog would be close to animals like cat. The average would not be close to other food like burgers or spaghetti. Another example is "*hong kong style food*". As it consists of many words, the averaged word vector could be far away from "*food*" in vector space.

To address this problem, inspired by He et al. (2017), we instead model each target as a mixture of K aspect embeddings where we would like each embedded aspect to represent a cluster of closely related targets. We use an *autoencoder* structure to learn both the aspect embeddings as well as the representation of the target as a weighted combination of the aspect embeddings. The weight vector represents the probability distribution over aspects for the given target. The autoencoder structure is jointly trained with a neural attention-based sentiment classifier to provide a good target representation as well as a high accuracy on the predicted sentiment. We found the learned embeddings to be semantically meaningful, i.e., embeddings of words that are semantically related appear close to the same aspect embedding. For example, embeddings of the words *service*, *servers*, *staff*, and *courteous* appear close to the same aspect embedding, which we interpret to represent the aspect *service*.

Our second approach exploits syntactic information to construct a syntax-based attention model. The attention models used in previous works give equal importance to all context words. In that case, the computed attention weights rely entirely on the semantic associations between context words and the target. However, this may not be sufficient for differentiating opinions words for different targets. Instead, our syntax-based attention mechanism selectively focuses on a small subset of context words that are close to the target on the syntactic path which is obtained by applying a dependency parser on the review sentence.

We conducted experiments on attention-based LSTM models using the SemEval 2014, 2015, and 2016 datasets. The results show that attention-based LSTM can be substantially improved by incorporating our two proposed methods, and that the resulting model outperforms all baseline methods on aspect-level sentiment classification.

2 Related Work

Under supervised learning conditions, aspect-level sentiment classification is typically considered as a classification problem. Early works (Boiy and Moens, 2009; Jiang et al., 2011; Kiritchenko et al., 2014; Wagner et al., 2014) mainly used manually designed features such as sentiment lexicon, n-grams, and dependency information. However, these methods highly depend on the quality of the designed features, which is labor-intensive. With the advances of deep learning methods, various neural models (Dong et al., 2014; Nguyen and Shirai, 2015; Vo and Zhang, 2015; Tang et al., 2016a; Tang et al., 2016b; Wang et al., 2016; Zhang et al., 2016; Liu and Zhang, 2017; Chen et al., 2017; He et al., 2018) have been proposed for automatically learning target-dependent sentence representations for classification. The main idea behind these works is to develop neural architectures that are capable of learning continuous features without feature engineering and at the same time capturing the intricate relatedness between a target and context words.

Among these works, attention-based neural models have attracted growing interest due to their ability to explicitly capture the importance of context words. Tang et al. (2016b) have shown that a better sentence representation could be obtained by stacking multiple layers of attention. In the work of Wang et al (2016), a variant of attention-based LSTM was proposed. Chen et al. (2017) also adopts multiple layers of attention and aggregates the attention outputs through a recurrent neural network.

As aspect information is very beneficial, in some works (Wang et al., 2016; Cheng et al., 2017; Ma et al., 2018), an aspect embedding is directly used to capture the importance of context words through an attention mechanism, where the authors assume that the aspect label is provided as an input. Unlike them, we do not assume that the aspects of each sentence are given. Instead, we propose to learn the probability distribution over aspects for the given target, and use the weighted summation of aspect embeddings for target representation. The probability distribution and the aspect embeddings are learned via an unsupervised objective, which is jointly trained with the neural attention-based sentiment classifier.

3 Model Description

We propose two approaches to improve the effectiveness of the attention mechanism. The approaches may be applied more generally but we use them on attention-based LSTM as it has been widely used in previous works for sentiment analysis (Chen et al., 2016; Wang et al., 2016; Chen et al., 2017; Liu and Zhang, 2017; Ma et al., 2018). We first give the task definition in (§3.1). Then, we briefly describe the architecture of attention-based LSTM (§3.2) and introduce the two proposed approaches (§3.3 & §3.4). Finally, we describe the overall architecture of our model for aspect-level sentiment classification and the training objective (§3.5).

3.1 Task Definition and Notation

Given a review sentence $s = (w_1, w_2, ..., w_n)$ consisting of n words, and an opinion target occurring in the sentence $a = (a_1, a_2, ..., a_m)$ consisting of a subsequence of m continuous words from s, aspect-level sentiment classification aims to determine the sentiment polarity of sentence s towards the opinion target a. When dealing with a text corpus, we begin by associating each word w with a continuous feature vector $\mathbf{e}_w \in \mathbb{R}^d$, also known as word embedding (Mikolov et al., 2013), where d denotes the embedding dimension. The vectors associated with the words correspond to the rows of a word embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size.

3.2 Attention-based LSTM

We briefly describe a conventional attention-based LSTM in this subsection. Given a sequence of word embeddings $\{\mathbf{e}_{w_1}, \mathbf{e}_{w_2}, ..., \mathbf{e}_{w_n}\}$ of a sentence s, LSTM with trainable parameters θ_{lstm} makes use of three gates to discard or pass the information through time (Hochreiter and Schmidhuber, 1997), and outputs a sequence of hidden vectors $h = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$. The sentence representation $\mathbf{z}_s$ used for sentiment classification is then computed as the weighted summation of hidden vectors.

$$\mathbf{z}_s = \sum_{i=1}^{n} p_i \mathbf{h}_i \tag{2}$$

A positive weight p_i is computed for each $\mathbf{h}_i$, which can be interpreted as the probability that w_i is the right word to focus on when inferring the sentiment polarity of the opinion target a. The value p_i is computed by an attention model, which conditions on the hidden vector $\mathbf{h}_i$ as well as the target representation. In previous works (Tang et al., 2016b; Wang et al., 2016; Liu and Zhang, 2017), the attention process is usually described with the following equations:

$$p_i = \frac{\exp(d_i)}{\sum_{j=1}^{n} \exp(d_j)} \tag{3}$$

$$d_i = f_{score}(\mathbf{h}_i, \mathbf{t}_s) \tag{4}$$

$$\mathbf{t}_s = \frac{1}{m} \sum_{i=1}^{m} \mathbf{e}_{a_i} \tag{5}$$

where f_{score} is a function that computes a score for word w_i according to the semantic association between $\mathbf{h}_i$ and $\mathbf{t}_s$, and $\mathbf{t}_s$ is the vector representation of the given target.

3.3 Target Representation

Most previous work (Tang et al., 2016b; Liu and Zhang, 2017; Chen et al., 2017) represent a target by averaging its component word or hidden vectors as shown in Equation 5. Simple averaging may not

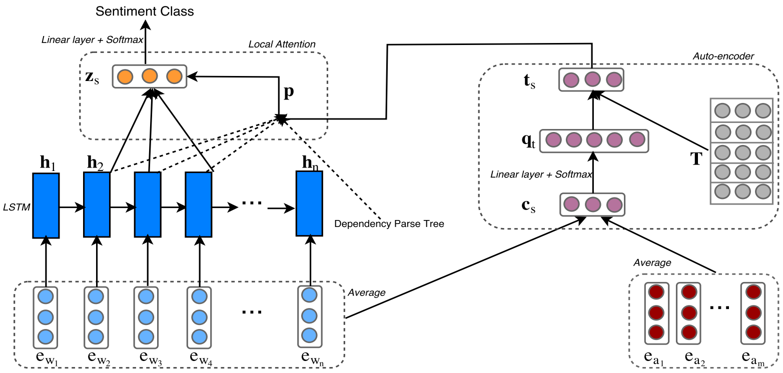


Figure 1: The overall architecture of the integrated model.

capture the real semantics of the target well. Inspired by He et al. (2017), we represent the target as a weighted summation of aspect embeddings, as illustrated in Fig. 1. An aspect embedding matrix is represented by $\mathbf{T} \in \mathbb{R}^{K \times d}$, where K, the number of aspects defined by the user, is much smaller than V. The process is formalized as follows:

$$\mathbf{t}_s = \mathbf{T}^\top \cdot \mathbf{q}_t \tag{6}$$

$$\mathbf{q}_t = \mathit{softmax}(\mathbf{W}_t \cdot \mathbf{c}_s + \mathbf{b}_t) \tag{7}$$

$$\mathbf{c}_s = \mathit{Average}\Big(\frac{1}{m}\sum_{i=1}^{m} \mathbf{e}_{a_i}, \frac{1}{n}\sum_{j=1}^{n} \mathbf{e}_{w_j}\Big) \tag{8}$$

where *Average* returns the mean of the input vectors. $\mathbf{c}_s$ captures both target information and context information. $\mathbf{q}_t$ is the weight vector over K aspect embeddings, where each weight represents the probability that the target belongs to the related aspect. $\mathbf{W}_t$ and $\mathbf{b}_t$ are a weight matrix and a bias vector respectively.

We would like the learned aspect embeddings to be meaningful and semantically coherent. This would allow us to interpret an aspect by looking at its nearby words in vector space. However, the aspect embedding matrix $\mathbf{T}$ is randomly initialized. It is difficult to obtain coherent aspect embeddings if we only rely on the training of the sentiment classifier. Therefore, we add an unsupervised objective function to ensure the quality of the aspect embeddings, which is jointly trained with the attention-based LSTM. Indeed, we can understand the process shown by Eq. (6) (7) (8) as an autoencoder, where we first reduce $\mathbf{c}_s$ from d dimensions to K dimensions with softmax non-linearity. Only the dimensions that are relevant to the aspects are retained in $\mathbf{q}_t$, whereas the other dimensions are removed. Then we reconstruct $\mathbf{c}_s$ from $\mathbf{q}_t$ through linear combination of aspect embeddings. The unsupervised objective is thus to minimize the reconstruction error as shown below:

$$U(\theta) = -\sum_{(s,a)\in D} \log(\min(\epsilon, \mathit{CosSim}(\mathbf{t}_s, \mathbf{c}_s))) \tag{9}$$

where cosine similarity *CosSim()* is used as the similarity measure. ϵ denotes a very small positive number. We set it to 10^{-7} in all experiments. D denotes all training samples, (s, a) denotes a sentence-target pair, and $\theta = \{\mathbf{E}, \mathbf{T}, \mathbf{W}_t, \mathbf{b}_t\}$ is the set of trainable parameters.

The learning process can also be viewed as multi-task learning where the unsupervised objective shown as Eq. 9 is an auxiliary task. Multi-task learning can help to reduce the amount of data required for learning and to improve the model generalization ability.

3.4 Syntax-based Attention Mechanism

The attention mechanism used in previous works (Tang et al., 2016b; Wang et al., 2016; Liu and Zhang, 2017) gives equal importance to all context words, where the attention weight is merely a measure of semantic association between the target and the context word. But intuitively not all words are equally important for determining the polarity of a target. Words that appear near the target or have a modifier relation to the target, for example, are more important and should receive higher weight. This is particularly true for opinion words that express sentiment and when there are multiple targets and multiple opinion words in one sentence. To address this issue, we propose an attention mechanism that also encodes the syntactic structure of a sentence, where syntactic information is obtained from a dependency parser. Fig. 2 shows the dependency tree of an example sentence. The opinion words that are closer to the target in the dependency tree are more relevant for determining its sentiment. In our model, we define the location l of a context word as its distance to the target[1] along the dependency path. The attention model selectively attends to a small window of context words based on their location. We use ws to denote the attention window size. In our experiment, we ignore context words whose location is larger than ws and for context words within the window, different weights are applied so that words closer to the target receive more attention. The details of the proposed syntax-based attention model are described as follows:

$$p_i = \frac{d_i}{\sum_j d_j} \tag{10}$$

$$d_i = \begin{cases} \frac{1}{2^{(l_i-1)}} \cdot exp(f_{score}(\mathbf{h}_i, \mathbf{t}_s))), & \text{if } l_i \in [1, ws] \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where t_s is the target representation constructed using the method described in §3.3. We adopt a simple score function as follows:

$$f_{score}(\mathbf{h}_i, \mathbf{t}_s) = tanh(\mathbf{h}_i^T \cdot \mathbf{W}_a \cdot \mathbf{t}_s) \tag{12}$$

where $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ is a trainable weight matrix.

3.5 Overall Architecture and Training Objective

After incorporating the two proposed approaches into the attention-based LSTM, our final model is illustrated in Fig. 1. The attention-based LSTM component is associated with the categorical cross entropy loss of sentiment classification. The loss function is given below:

$$J(\theta) = -\sum_{(s,a)\in D} \sum_{c \in C} P^g_{(s,a)}(c) \log(P_{(s,a)}(c)) \tag{13}$$

where C is the collection of sentiment classes, $P^g_{(s,a)}(c)$ is either 1 or 0, indicating whether the gold label is c for (s, a), and $P_{(s,a)}(c)$ is the predicted probability that (s, a) has sentiment class c. $\theta = \{\mathbf{E}, \mathbf{T}, \mathbf{W}_t, \mathbf{b}_t, \mathbf{W}_a, \theta_{lstm}\}$ is the set of trainable parameters.

The aspect embeddings in $\mathbf{T}$ may become similar to each other during training. To ensure diversity, we employ a regularization term to enforce the uniqueness of each aspect embedding:

$$R(\theta) = \|(\mathbf{T}_{norm} \cdot \mathbf{T}_{norm}^\top - \mathbf{I})^2\| \tag{14}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{T}_{norm}$ is the L_2 normalization of $\mathbf{T}$, and $\|\|$ denotes the sum of all entries in the matrix. R reaches the minimum when the dot product between any two different aspect embeddings is zero. Thus, the regularization term aims to enforce orthogonality among the rows of $\mathbf{T}$, which punishes redundancy between aspect embeddings.

The final objective function of our model is defined as:

$$L(\theta) = J(\theta) + \lambda_u U(\theta) + \lambda_r R(\theta) \tag{15}$$

where λ_u and λ_r are hyperparameters that control the weights of the unsupervised objective described in §3.3 and the regularization term respectively.

[1] For a target containing multiple words, the distance between a context word and each word in the target is computed, and the minimal value is used to define the location of the context word.

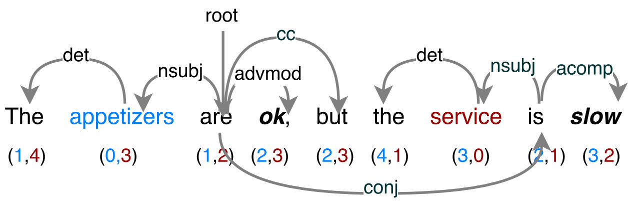


Figure 2: A dependency tree example. The numbers indicate the distances from the word to the two targets respectively along the syntactic path.

	Dataset	Pos	Neg	Neu
D1	Restaurant14-Train	2164	807	637
	Restaurant14-Test	728	196	196
D2	Laptop14-Train	994	870	464
	Laptop14-Test	341	128	169
D3	Restaurant15-Train	1178	382	50
	Restaurant15-Test	439	328	35
D4	Restaurant16-Train	1620	709	88
	Restaurant16-Test	597	190	38

Table 1: Dataset description.

4 Experiments

4.1 Datasets

We evaluate our proposed model on four benchmark datasets, taken from SemEval 2014 task 4 (Pontiki et al., 2014), SemEval 2015 task 12 (Pontiki et al., 2015), and SemEval 2016 task 5[2] (Pontiki et al., 2016). Each training and test sample in the 2014 datasets consists of the review sentence, the opinion target, and the sentiment polarity towards the opinion target. Following previous works (Tang et al., 2016b; Wang et al., 2016), we remove samples with *conflicting* polarity in the 2014 datasets – the number of samples in that class is very small and incorporating it will make the training dataset extremely unbalanced. The data format in the 2015 and 2016 datasets is a bit different, where each opinion target is also associated with one or multiple aspects and thus can have multiple sentiment polarities. Below is an example:

The food was delicious but expensive.
(target=“food”, aspect=food#quality, polarity=Pos)
(target=“food”, aspect=food#prices, polarity=Neg)

Since our model only takes a sentence and an opinion target as input, without using the aspect information, we remove a sample in both training and test sets if the opinion target has different polarities as the example above. This removes about 5% and 4% of test samples from the 2015 and 2016 datasets respectively. Statistics of the resulting datasets are presented in Table 1.

We initialize word embeddings using the 300-dimension GloVe vectors supplied by Pennington et al. (2014) and we use the dependency parser from spaCy[3] to obtain dependency paths of review sentences. We randomly select 20% of the original training data as the development set and only use the remaining 80% for training. Values for the hyperparameters are obtained empirically on the development set of one task and are fixed for all other experiments. The dimension of the LSTM hidden vectors is set to 300, the objective weights λ_u and λ_r are set to 1 and 0.1 respectively, the attention window size ws is set to 5 and the number of aspects K is set to 8.

We use RMSProp with base learning rate set to 0.001 and decay rate set to 0.9 for network training. The minibatch size is set to 32. As a regularizer, we apply dropout (Srivastava et al., 2014) with probability 0.5 to the LSTM layer and the output layer. We train the network for a fix number of epochs and select the best model according to the performance on the development set, and evaluate it on the test set.

4.2 Model Comparisons

We compare our model with the following baselines:

(1) **Feature-based SVM** (Kiritchenko et al., 2014): We compare with the reported results of a top system in SemEval 2014. We could not directly compare with the reported results from SemEval 2015 and 2016 as their model inputs are different from ours (aspect is also one of their inputs).

(2) **LSTM**: An LSTM network is built on top of word embeddings. The mean over hidden vectors is used as the sentence representation.

[2] Although there is another English dataset in the laptop domain from SemEval 2015 and 2016, it does not contain opinion targets, thus it cannot be used directly in our work.

[3] https://spacy.io

Methods	D1		D2		D3		D4	
	Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1
Feature-based SVM	80.16	NA	70.49	NA	NA	NA	NA	NA
LSTM	75.23	64.21	66.79	64.02	75.28	54.10	81.94	58.11
LSTM+ATT	76.83	66.48	68.07	65.27	77.38	60.52	82.73	59.12
TDLSTM	75.37	64.51	68.25	65.96	76.39	58.70	82.16	54.21
TDLSTM+ATT	75.66	65.23	67.82	64.37	77.10	59.46	83.11	57.53
ATAE-LSTM	78.60	67.02	68.88	65.93	78.48	62.84	83.77	61.71
MM	76.87	66.40	68.91	63.95	77.89	59.52	83.04	57.91
RAM	78.48	68.54	72.08	68.43	79.98	60.57	83.88	62.14
Ours: LSTM+ATT+TarRep	78.95	68.67	70.69	66.59	80.05	**68.73**	84.24	**68.62**
Ours: LSTM+SynATT	80.45	71.26	**72.57**	69.13	80.28	65.46	83.39	66.83
Ours: LSTM+SynATT+TarRep	**80.63***	**71.32***	71.94	**69.23**	**81.67***	66.05*	**84.61***	67.45*

Table 2: Average accuracies and Macro-F1 scores over 5 runs with random initializations. The best results are in bold. * indicates that our full model (LSTM+SynATT+TarRep) is significantly better than LSTM, LSTM+ATT, TDLSTM, TDLSTM+ATT, ATAE-LSTM, MM and RAM with $p < 0.05$ based on one-tailed unpaired t-test.

(3) **LSTM+ATT**: The model described in section 3.2.

(4) **TDLSTM** (Tang et al., 2016a): It uses a forward LSTM and a backward LSTM to encode the information before and after the target.

(5) **TDLSTM+ATT**: It extends TDLSTM by incorporating an attention mechanism.

(6) **ATAE-LSTM** (Wang et al., 2016): It is a variant of the attention-based LSTM model.

(7) **MM** (Tang et al., 2016b): It uses multi-hops of attention layers for sentence representation.

(8) **RAM** (Chen et al., 2017): It uses multi-hops of attention layers and combines the multiple attention outputs with a recurrent neural network for sentence representation.

We produce the results of TDLSTM, ATAE-LSTM, and MM with the source codes released by their authors. We re-implement RAM following the instructions in its paper as the code is not available. The comparison results are shown in Table 2. Both accuracy and macro-F1 are used for evaluation as the label distributions are unbalanced. The reported numbers are obtained as the average value over 5 different runs with random initializations for each method. Significant test results are included for testing the robustness of methods under random parameter initializations. We also show the effect of each proposed approach: **LSTM+ATT+TarRep** denotes the model where the proposed target representation is used while the attention model remains the same as LSTM+ATT; **LSTM+SynATT** denotes the model where only the conventional attention is replaced by our syntax-based attention; and **LSTM+SynATT+TarRep** denotes the full model with both approaches integrated as shown in Fig. 1.

We make the following observations: 1) Feature-based SVM is still a strong baseline, our best model achieves competitive results on D1 and D2 without relying on so many manually-designed features and external resources. 2) Compared with all other neural baselines, our full model achieves statistically significant improvements ($p < 0.05$) on both accuracies and macro-F1 scores for D1, D3, D4. 3) Compared with LSTM+ATT, all three settings of our model are able to achieve statistically significant improvements ($p < 0.05$) on all datasets. This demonstrates that both proposed approaches are effective. 4) The integrated full model overall achieves the best performance compared to using only one of the two proposed approaches. This indicates that the two proposed approaches are complementary, thus further improvements could be obtained when combining them. 5) The proposed target representation is more helpful on restaurant domain (D1, D3, and D4) than laptop domain (D2). A plausible reason is that restaurant domain has clearer aspects for opinion targets, while it is much harder to determine the aspects for many opinion targets in the laptop domain. Since our model represents the target as weighted summation of aspect embeddings, domains with clear aspects may benefit more from the model.

4.3 Model Analysis

We conduct more detailed analysis of the proposed approaches quantitatively and qualitatively. By examining the final test outputs of the relevant models, we try to investigate what kind of errors made by the baseline can be more effectively treated by our proposed approaches.

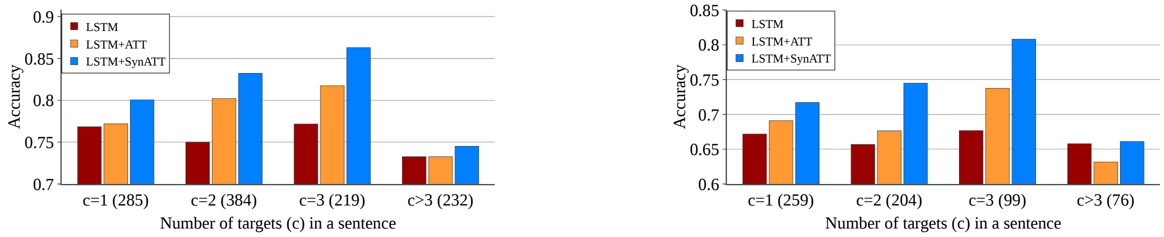
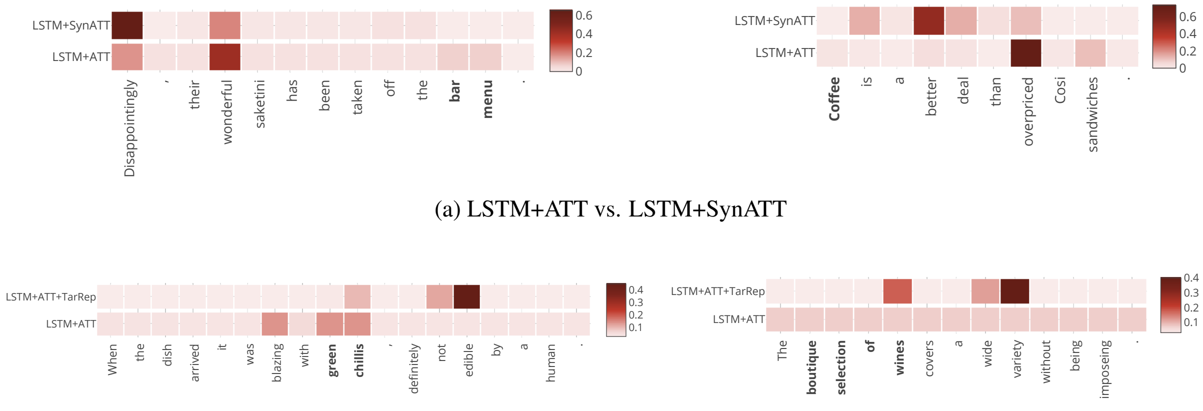


Figure 3: Classification accuracies on groups with different number of opinion targets. The number in brackets indicates the number of test instances in that group. Restaurant14 (left), Laptop14 (right).



(a) LSTM+ATT vs. LSTM+SynATT

(b) LSTM+ATT vs. LSTM+ATT+TarRep

Figure 4: Attention visualization on example sentence-target pairs. The opinion target is in bold.

Impact of Syntax-based Attention Syntax-based attention is supposed to better differentiate opinion contexts for different targets when there are multiple targets appearing in the sentence. To verify this, we compare LSTM+SynATT with LSTM and LSTM+ATT on sentences grouped by their number of targets. Fig. 3 shows the accuracies on the test sets of Restaurant14 (D1) and Laptop14 (D2).

LSTM+SynATT performs the best on all groups. In particular, it performs substantially better on groups with two or three targets. By analyzing a number of examples from these groups, we find that the proposed syntax-based attention is more effective in capturing the relevant opinion context for a given target when there are multiple targets in the sentence. Two examples are given in Fig. 4a, where our syntax-based attention successfully captures the correct opinion word towards the target of interest, whereas since conventional attention only relies on semantic association between words and the target, it fails by mis-attending to the opinion word towards other target which has similar aspect semantics.

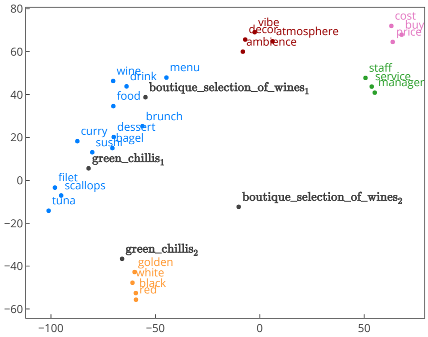


Figure 5: t-SNE visualization of embedding space. Subscripts $_1$ and $_2$ are used to denotes the target representation learned by our method and the method of averaging word vectors used in previous works respectively.

In addition, we observe that all models perform poorly on the group with more than three targets. By analyzing the errors, we find two main causes. First, those sentences are relatively long, involving more complex opinion expressions and sentence structures. Second, the proportion of neutral samples

service	table	atmosphere	price	champagne	curry	bagels	scallops
servers	tables	ambience	prices	wine	thai	bagel	fillet
staff	reservation	ambiance	buy	drink	dumplings	dessert	mignon
courteous	reservations	decor	buying	bottle	sushi	pastries	salmon
waitstaff	waiting	surroundings	cost	wines	samosa	pies	tuna

Table 3: Top 5 representative words of the eight discovered aspects on Restaurant14

contained in this group is much higher than in other groups. Since the number of neutral samples in the training set is small, the trained classifier has difficulties in predicting neutral samples in the test set.

Impact of Target Representation To investigate how the proposed target representation helps to improve performance, we extract test examples from Restaurant14 which are mis-classified by LSTM+ATT but are correctly classified by LSTM+ATT+TarRep. Among these examples, 56% are associated with opinion targets consisting of more than one words. Two examples are shown in Fig. 4b where the targets are "green chillis" and "boutique selection of wines" respectively. Fig. 5 uses t-SNE visualization to show the comparison of the learned target representations between our method and the method of averaging word vectors used in previous works on these two examples. In Fig. 5, we can observe that simply averaging the component word vectors fails to capture the correct semantics of both targets, as the target representations are far away from the food-related words in the embedding space. Due to the inaccurate representation of target, as shown in Fig. 4b, LSTM+ATT fails to attend to the right opinion context in both examples. Our proposed target representation is able to capture the correct aspect semantics for both targets and as a result, the attention mechanism can capture the correct opinion context.

Furthermore, the proposed target representation also outputs aspect embeddings after the training process. Each aspect can be interpreted by its nearby words in vector space. Table 3 presents top representative words of the eight discovered aspects on Restaurant14. The words are ranked based on their cosine similarities with the aspect embedding. As shown, each aspect is semantically coherent and our model is able to discover the typical aspects of a restaurant such as food, ambience, service, and price. Since q_t in Equation (8) represents the probability distribution over aspects for the input target, our model could additionally be used to map the input target to an aspect. We did not conduct further experiments on this since it is not our main focus in this work, but it could be an interesting direction to explore in future.

4.4 Remaining Error Analysis

We additionally conduct a careful analysis of a subset of errors made by our full model, in order to better understand its limitations. To do that, we randomly sample 100 examples with classification errors on the test set of Restaurant14, and classify them into several error categories. Table 4 shows the top three error categories, the corresponding proportions, and some representative examples for each category. The top category is *Neutral*, which denotes examples where the gold sentiment label is neutral. There are two main groups of errors under this category: (1) The polarity of the target is affected by other sentiment words in the sentence. As shown in example 1), the sentence holds a positive sentiment on atmosphere, but expresses no specific opinion on *drinks*. However, affected by the word *perfect*, the predicted sentiment towards *drinks* is positive. Although the proposed attention mechanism aims to address this type of errors, it still fails on complex examples; (2) The sentence is objective, with no opinion expression such as example 2). Since there are many more positive training examples, the predictions on such neutral examples are often biased towards positive sentiment.

The second most common error category is *Complex*, which includes examples with implicit opinion expressions (example 3) or those that require deep comprehension to be understood (example 4). This type of errors is difficult to handle with current techniques, especially when trying to build an end-to-end neural network. The diversity and low frequency of those expressions make it hard for a statistical approach to capture their patterns. For errors made on examples with negation words, we believe this is due to the insufficient training data such that LSTM cannot effectively capture certain sequential patterns.

No.	Category	(%)	Examples
1	Neutral	43	1) *A beautiful atmosphere, perfect for [drinks]$_{neu}$* 2) *We started with the [scallops]$_{neu}$ and [asparagus]$_{neu}$ and also had the [soft shell crab]$_{neu}$.*
2	Complex	28	3) *The [banana pudding]$_{neg}$ they serve has never seen as oven ...* 4) I can understand the [prices]$_{neg}$ if it served better food.
3	Negation words (sentiment shifter)	9	5) *I thought the [food]$_{neg}$ is not cheap at all compared to Chinatown.* 6) *The [dinner]$_{pos}$ here is never disappointing.*

Table 4: Top three error categories.

5 Conclusion

We propose two novel approaches to improve the effectiveness of attention mechanism for aspect-level sentiment classification. In our experiments, we show quantitatively and qualitatively that both methods help to improve the performance of a conventional attention-based LSTM. The integrated model achieves the best results over baseline methods.

As future work, we can consider improving the accuracy on neutral examples. Possible methods include data augmentation on neutral examples and integration of linguistic knowledge to better determine target-relevant opinion expressions.

References

Erik Boiy and Marie-Francine Moens. 2009. A machine learning approach to sentiment analysis in multilingual web texts. *Information Retrieval*, 12:526–558.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

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 2017)*.

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *International Conference on Information and Knowledge Management (CIKM 2017)*.

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 2014)*.

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 2017)*.

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 2018)*.

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

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 (ACL 2011)*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *International Workshop on Semantic Evaluation (SemEval 2014)*.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*.

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 (AAAI 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS, 2013)*.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval 2014)*.

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 2015)*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud Maria Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval 2016)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for target-dependent sentiment classification. In *International Conference on Computational Linguistics (COLING 2016)*.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent Twitter sentiment classification with rich automatic features. In *International Joint Conference on Artificial Intelligence (IJCAI 2015)*.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. DCU: Aspect-based polarity classification for SemEval task 4. In *International Workshop on Semantic Evaluation (SemEval 2014)*.

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 2016)*.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *AAAI Conference on Artificial Intelligence (AAAI 2016)*.