The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Latent Opinions for Aspect-Level Sentiment Classification

Bailin Wang
College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01003, USA
bailinwang@cs.umass.edu

Wei Lu
Singapore University of Technology and Design
8 Somapah Road
Singapore, 487372
luwei@sutd.edu.sg

Abstract

Aspect-level sentiment classification aims at detecting the sentiment expressed towards a particular target in a sentence. Based on the observation that the sentiment polarity is often related to specific spans in the given sentence, it is possible to make use of such information for better classification. On the other hand, such information can also serve as justifications associated with the predictions. We propose a segmentation attention based LSTM model which can effectively capture the structural dependencies between the target and the sentiment expressions with a linear-chain conditional random field (CRF) layer. The model simulates human's process of inferring sentiment information when reading: when given a target, humans tend to search for surrounding relevant text spans in the sentence before making an informed decision on the underlying sentiment information. We perform sentiment classification tasks on publicly available datasets on online reviews across different languages from SemEval tasks and social comments from Twitter. Extensive experiments show that our model achieves the state-of-the-art performance while extracting interpretable sentiment expressions.

Introduction

Aspect-level sentiment analysis (Pang, Lee, and others 2008; Liu 2012) has been popular recently both in academic communities and industry since it allows the detailed examination of the user-generated text. Several subtasks are defined to achieve this goal, e.g., extraction of opinion targets, detection of aspect categories, and targeted sentiment classification (Pontiki et al. 2014). Aspect-level sentiment classification is one of these subtasks that aims at detecting the sentiment expressed towards a particular target appearing in a given sentence. Regarding such a task as a simple sentiment classification problem at the sentence level (Socher et al. 2013) is undesirable since different targets in the same sentence may have different sentiment information. Figure 1 shows a concrete example. One key observation that we can make here is that typically there is always one or more opinion expressions that contribute to the sentiment of the target. Unlike the joint models (Li et al. 2010; Zhao et al. 2010; Wang et al. 2016a) which extract the opinion terms explicitly, attention-based models (Wang et al. 2016b) regard the opinions as latent variables and learns to focus on different parts of the sentence given a particular target. However, standard attention mechanism does not model structural dependencies that exist in sentences. For example, as we can see from the example given in Figure 1, the opinion expression associated with a target may be in the form of a chunk or a linear span structure. In general, the opinion expressions can consist of multiple spans of texts. Accurately modeling and extracting such structural information can be extremely crucial especially when the sentences contain multiple targets.

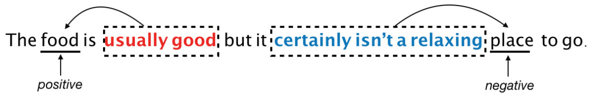
The food is usually good but it certainly isn't a relaxing place to go.
positive
negative

Figure 1: An example of a review with two target terms which have different sentiments. The opinions are highlighted with a box and point to their corresponding targets.

Therefore, we propose a model to capture such structural information so that corresponding opinions can be identified to facilitate the sentiment classification. Specifically, we incorporate a layer that is analogous to conditional random field (CRF) (Lafferty, McCallum, and Pereira 2001) in the attention modeling process to capture the structural dependencies. The resulting new attention mechanism will be able to perform soft selections of opinion expressions in the form of word spans. This can be viewed as an extension of the standard attention mechanism. We call such a novel attention mechanism *segmentation attention* in this paper.

From the modeling perspective, standard attention mechanism that is widely used in the community can be viewed as the process of performing soft selections of individual words independently, whereas our segmentation attention mechanism essentially captures the dependencies between adjacent words in the process. As a result, we can understand the mechanism as a process that performs soft selections of a consecutive sequence of words or spans. This is based on the observation that it is usually a coherent opinion span rather than individual words scattered in the sentence that form meaningful information, contributing towards the sentiment. Standard attention-based models may make the wrong decision when complex expressions are involved. To bring positional information to the attention model, previous methods either add the positional embeddings to the input (Tang, Qin, and Liu 2016) or use position-based reweighting (Chen et al. 2017). Instead of directly building connections between the position of a context word and its influence on the sentiment, we want to build a structural model that is expected to learn this kind of correlation. Deep models such as Recurrent Neural Networks (RNN) may be able to capture the structural information implicitly. However, learning such information automatically from data might require a substantial amount of data. We believe developing models that can explicitly capture the structural bias in the attention modeling process is still useful and important.

Another advantage that our model brings is that we can extract the opinion expressions explicitly during its decoding step and these opinions can serve as justifications of our predictions. This property makes our model significantly more interpretable compared with existing methods and allows us to have a better understanding of the underlying predictions associated with the neural networks. Furthermore, we introduce two additional regularizers to guide the model to learn meaningful opinions. Specifically, as we have mentioned, the opinions that contribute to the sentiment of a target are usually short and coherent spans rather that disconnected words. With the guidance of the additional regularizers, the segmentation attention layer is expected to select a consecutive sequence of words as opinions.

Our model basically consists of two components: A bidirectional long short-term memory network (BiLSTM) (Graves, Mohamed, and Hinton 2013) layer that runs through the words in the sentence sequentially to get contextual information for each word, and segmentation attention layer that aims to distill the sentiment information from the sentence. We believe that with these components, the model is capable of learning phrase-like features and generate reasonable spans as opinions.

Experiments are conducted in two stages to verify the effectiveness of our proposed model from different aspects. First, we evaluate our model on three English datasets which consist of online reviews and social comments. We show the effectiveness of BiLSTM and segmentation attention by using some basic variants which exclude such an attention mechanism. As our model extracts opinions without explicit supervision, to understand the quality of the extracted opinions, we then conduct some qualitative analysis by comparing the extracted opinions with the manually annotated opinions. Second, we apply our model on additional review datasets across seven languages to examine the model's language sensitivity. Note that the datasets we are using at this stage are specifically from SemEval 2016 Task 5. It also provides aspect category like "food quality" coupled with each target term, which reveals more information about the target.

The main contributions of this work include:

- We propose a novel segmentation attention based model for aspect sentiment analysis, which can capture structural dependencies between a target and its opinions, as well as dependencies between opinion words.
- Extensive experiments are conducted on standard datasets consisting of online reviews and social comments. Results show that our model consistently achieves the state-of-art performance on the aspect sentiment analysis task.
- We conduct comprehensive analyses to understand how our model works with the help of segmentation attention, including evaluation of latent extracted opinions.

Our implementation is available at https://github.com/berlino/SA-Sent

Related Work

Earlier approaches to sentiment analysis include rule-based methods (Ding, Liu, and Yu 2008) and SVM-based methods (Kiritchenko et al. 2014; Jiang et al. 2011), which usually require significant manual feature engineering efforts. Since Neural Networks (NN) have emerged as powerful approaches for sentiment analysis (Kim 2014; Socher et al. 2013), several methods are adapted for this fine-grained task. AdaRNN (Dong et al. 2014) tries to propagate the sentiment of words to the target using recursive neural networks given the syntactic tree of the sentence. However, the underlying assumption on the availability of the syntactic tree may not always hold especially when informal text such as online comments and reviews are considered. Chen et al. (2016) use convolutional neural networks (CNN) to infer the sentiment of a target by identifying the sentiment of the clause in which the target lies. Tang et al. (2016) uses two LSTM networks running towards target word from left and right respectively to capture the contextual information of a target.

There are also different kinds of joint models for this task. In opinion extraction, one can refine the opinion labels by adding sentiment polarity (such as positive opinion) (Li et al. 2010) so that both opinion expressions and the polarity information can be jointly captured using sequence labeling models. Also, target extraction task (Zhao et al. 2010; Wang et al. 2016a) can be added to this joint learning framework. However, the annotation of opinion expressions is quite expensive. Mitchell et al. (2013) takes a novel way to extract the sentiment target together with its sentiment polarity based on the assumption that surrounding context reveals enough information to detect the target's sentiment. Li and Lu (2017) takes a latent-variable approach that learns the latent sentiment scope which contains both target and opinion expressions. However, it cannot handle the cases where different targets share the same opinion expression.

The attention mechanism has proven to be very effective in machine translation (Bahdanau, Cho, and Bengio 2015), question answering (Sukhbaatar et al. 2015) and many other tasks. Our model is inspired by structural attention network (Kim et al. 2017) which extends the standard attention to directly model structural dependencies between source elements. In contrast with the soft attention mechanism, "hard" (stochastic) attention (Xu et al. 2015) is also leveraged by (Lei, Barzilay, and Jaakkola 2016) which uses a separate model to generate rationales for neural predictions. Yu, Lee, and Le (2017) propose a model that can skip the irrelevant text in a "hard" way. These methods share the same spirit with our model, but they usually require methods like policy gradient for training.

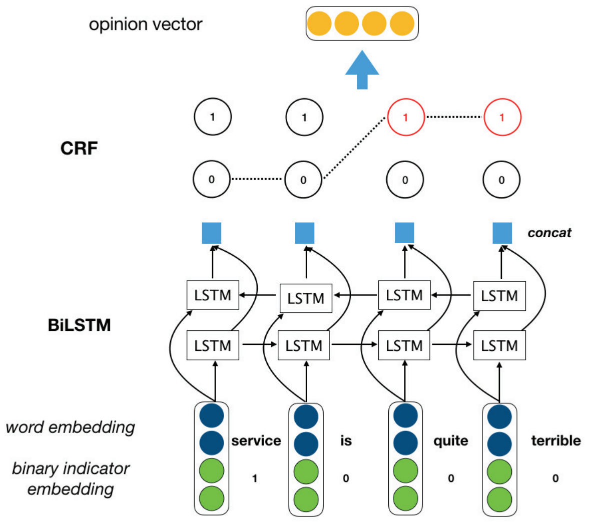

Figure 2: Model architecture. Target term "service" is indicated by embeddings of a binary indicator which are depicted in green. The model encodes the contextual information for each word using BiLSTM. Based on its hidden states, the internal segmentation attention layer aims at selecting the opinion words of interest. The nodes shown in red imply that their corresponding words have the most impact on the classification of the target's sentiment.

Model

The problem can be formulated as follows. Given a sentence with n words $\{w_1...w_i, w_{i+1}, ..., w_j, w_{j+1}, ..., w_n\}$ and a target $\{w_i, ..., w_j\}$ which is a span in the sentence, we need to predict the sentiment polarity of the specific target.

The architecture is shown in Figure 2. Next, we will introduce all components sequentially from bottom to top.

Input Layer

First, the words are mapped to their vector representations by looking up an embedding table. Unlike conventional methods (Wang et al. 2016b; Chen et al. 2017) that directly encode the target with word embeddings, we use a binary feature to indicate whether each word is part of a target or not since the positional information of the target can be thereby encoded by the indicator sequence. Following a similar idea in (He et al. 2017), each binary indicator is mapped to a vector representation using a randomly initialized embedding table.

As we can see from Figure 2, the representation of each word is formed by concatenating the word's embedding and the binary feature embedding:

$$\mathbf{v}_t = [\mathbf{W_{emb}}(w_t), \mathbf{W_{mask}}(t \in [i,j])] \tag{1}$$

where $t \in [i,j]$ is a binary function indicating whether t-th word belongs to the target span $[i,j]$. $\mathbf{W_{emb}} \in \mathbb{R}^{d_1 \times |V|}$ and $\mathbf{W_{mask}} \in \mathbb{R}^{d_2 \times 2}$ are two matrices, where $|V|$ is the vocabulary size, and d_1 and d_2 are dimensions of word embedding and binary feature embedding respectively. If the aspect term is given, we also encode them using learnable vectors which are concatenated to the representation above.

BiLSTM Layer

A BiLSTM is then used to capture the contextual information for each word. The forward LSTM is computed as follows:

$$i_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{b}_i) \tag{2}$$

$$o_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{b}_o) \tag{3}$$

$$f_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{b}_f) \tag{4}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{b}_c) \tag{5}$$

$$\mathbf{c}_t = i_t \odot \tilde{\mathbf{c}}_t + f_t \odot \mathbf{c}_{t-1} \tag{6}$$

$$\mathbf{h}_t = o_t \odot \tanh(\mathbf{c}_t) \tag{7}$$

where $\mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_f, \mathbf{W}_c \in \mathbb{R}^{d_h \times (d_1+d_2+d_h)}$ are weight matrices used for different gates, $\mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_f, \mathbf{b}_c \in \mathbb{R}^{d_h}$ are bias vectors and $d_1 + d_2$ and d_h indicate the dimension of input vector and hidden state for LSTM respectively. $\odot$ denotes element-wise multiplication and σ stands for the sigmoid function.

The backward LSTM is very similar to the forward one except that the input sequence is fed in a reversed way. We concatenate the hidden states of both forward and backward LSTM to form the final representation. Note that we include the target information as input in Equation 1, so each $\mathbf{r}_i$ can be viewed as a target-specific representation for the word at position i in the given sentence.

$$\mathbf{r}_t = [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t] \tag{8}$$

where $[\cdot,\cdot]$ refers to the operation that concatenates two column vectors to form a single column vector. We use $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$ to denote all the word representations generated for the sentence.

Segmentation Attention

In order to directly model the structural dependencies between a target and its opinions (also between opinion words), we incorporate a segmentation attention layer to capture them.

As shown in Figure 2, we introduce a latent binary variable $z \in \{0,1\}$ for each word. This variable indicates whether its corresponding word is part of an opinion expression or not. Specifically we incorporate linear-chain CRF to specify the structural dependencies between these latent variables. Formally, the distribution of a selected sequence is parameterized as follows:

$$p(\mathbf{z}|\mathbf{R}) = \frac{1}{Z(\mathbf{R})} \prod_{c \in C} \psi(\mathbf{z_c}, \mathbf{R}) \tag{9}$$

$$Z(\mathbf{R}) = \sum_{\mathbf{z}'} \prod_{c \in C} \psi(\mathbf{z}'_\mathbf{c}, \mathbf{R}) \tag{10}$$

where $\mathbf{z}$ is formed by $\mathbf{z}_c$, each is defined over an individual clique c, and $\psi(\mathbf{z_c}, \mathbf{R})$ is the potential function of the clique c and $Z(\mathbf{R})$ is the partition function. Typically we define two kinds of potentials on the vertices and edges for such an undirected graphical model respectively.

$$\prod_{c\in C}\psi(\mathbf{z}_c|\mathbf{R})=\prod_{i=1}^{n}\psi_1(z_i|\mathbf{R})\prod_{i=1}^{n-1}\psi_2(z_i,z_{i+1}|\mathbf{R}) \quad (11)$$

where

$$\psi_1(z_i|\mathbf{R})=\exp(\mathbf{W}^v_{z_i}\cdot\mathbf{r}_i+b) \quad (12)$$

$$\psi_2(z_i,z_{i+1}|\mathbf{R})=\exp(\mathbf{W}^e_{z_i z_{i+1}}) \quad (13)$$

Here $\mathbf{W}^v\in\mathbb{R}^{2\times 2d_h}$ maps context representation to the feature score of each latent state, $\mathbf{W}^e\in\mathbb{R}^{2\times 2}$ is a transition matrix defined for each pair of latent state.

Feature Function Our purpose is to generate the representation of the latent opinions which we referred to as *opinion vector* in Figure 2, based on previous selections. Equation 15 gives a general form of the function to compute this value.

$$\mathbf{z}=[z_1,...,z_n] \quad (14)$$

$$\mathbf{m}=\sum_{\mathbf{z}}p(\mathbf{z})g(\mathbf{R},\mathbf{z}) \quad (15)$$

where $g(\mathbf{R},\mathbf{z})$ is a feature function that is defined based on the selection of opinions, and $\mathbf{m}$ is the expectation of this feature function. However, enumerating all the possible selections is computationally expensive. To make the procedure tractable, we only define the features on each vertex so that this function can be computed based on marginal probabilities which can be calculated efficiently using dynamic programming with a message passing or forward-backward style algorithm:

$$g(\mathbf{R},\mathbf{z})=\sum_{i=1}^{n}\mathbb{1}(z_i=1)\mathbf{r_i} \quad (16)$$

Given Equation 15 and 16, the opinion vector can be simplified into the following form.

$$\mathbf{m}=\sum_{i}p(z_i=1)\mathbf{r_i} \quad (17)$$

We can intuitively see from this Equation 17 that the segmentation attention layer is essentially distilling the information from the sentence that contributes to sentiment polarity of the target. During the model prediction, we extract the latent opinions explicitly and exactly using a Viterbi style decoding algorithm. These opinions extracted in an unsupervised manner give us the chance to further analyze and evaluate our model.

Regularizers In initial experiments, we observe that the common errors that such an attention-based model made are that they tend to focus on sentiment words even if these words are not semantically associated with the target. Based on the assumption that opinion expressions are usually short and coherent spans rather than disconnected sentiment words, we introduce two additional regularizers to guide the model.

There are basically two states for the hidden variable z: being a part of the opinion or not. Based on the observation that only a few opinion spans actually have the effect on the target's sentiment, frequent transitions between different states should be discouraged. This gives rise to our first regularizer below, which tries to encourage the state to stay the same:

$$\Omega_1(\mathbf{z})=\sum_i\sum_{j\neq i}\max(0,\mathbf{W}^e_{ij}-\mathbf{W}^e_{ii}) \quad (18)$$

Specifically, it enforces the transition feature value between different states to be smaller than the one between the same state. Otherwise, the model will get a penalty.

The second regularizer tries to enforce the model to attend to short and few spans that really matter:

$$\Omega_2(\mathbf{z})=\sum_{i=1}^{n}p(z_i=1) \quad (19)$$

Essentially, these regularizers bring some structural bias so that the model can focus on short yet meaningful opinion spans through segmentation attention layer.

Objective Function

The distribution of the sentiment tags is computed using softmax:

$$p(y_i|\mathbf{m})=\mathrm{soft\,max}(\mathbf{W_{tag}}\cdot\mathbf{m}+b_{tag}) \quad (20)$$

where $\mathbf{W_{tag}}\in\mathbb{R}^{2d_h}$ maps the opinion vector $\mathbf{m}$ to the feature score for each sentiment label and b_{tag} is a bias term.

We mainly focus on two kinds of models: the vanilla segmentation attention based LSTM (SA-LSTM) and the version augmented with the additional regularizers (SA-LSTM-P). In the training mode of mini-batch, loss function for SA-LSTM-P is defined as follows:

$$\mathcal{L}=\frac{1}{N}\left[\sum_{i=0}^{N}-y_i\log p(y_i)+\lambda_1\Omega_1(\mathbf{z})+\lambda_2\Omega_2(\mathbf{z})\right] \quad (21)$$

where λ_1 and λ_2 denotes the coefficient for each regularizer. Without these penalty terms, it becomes the objective function of SA-LSTM.

Experiments

Datasets

We primarily conduct two sets of experiments using two groups of datasets. The first group is used to analyze each component of our model in detail. In the second group, we focus on examining the model's language sensitivity issue.

The first group consists of review datasets from SemEval-2014 task 4 (Pontiki et al. 2014) and Twitter comments collected by (Dong et al. 2014). SemEval-2014 task 4 contains reviews from two domains: restaurant and laptop. Following previous work (Tang, Qin, and Liu 2016; Chen et al. 2017), as part of preprocessing, we also discard the sentences that contain "conflict" labels (where different sentiments are expressed towards the same aspect). In order to evaluate the latent opinions that the model learned, we also make use of the additional annotations for these two datasets from (Wang et al. 2016a) which include manually annotated labels for

Dataset	Laptop	Restaurant	Twitter	Restaurant *						Hotel *
	English	English	English	English	Spanish	French	Turkish	Russia	Dutch	Arabic
Train	2,313	3,602	6,257	2,507	2,720	2,530	1,535	4,022	1,711	10,509
Test	638	1,120	694	859	1,072	954	159	1,300	613	2,604

Table 1: Numbers of ⟨ sentence, target, polarity ⟩ tuples in each domain. Datasets with "*" are from SemEval 2016 Task 5.

opinion terms. The Twitter dataset is collected by querying Twitter API using entity words, so each sentence is usually paired with only one target. In the second group, we use restaurant and hotel reviews from SemEval-2016 task 5 involving seven different languages. All statistics are shown in Table 1. [1]

Comparison Methods

- **SVM** (Kiritchenko et al. 2014): The best method reported in SemEval 2014 task 4, which takes lexicon features, surface features and parsing features for SVM.
- **AdaRNN** (Dong et al. 2014): This model learns to adaptively propagate the sentiments of context words to the target node based on the dependency tree converted for the target with Recursive NNs.
- **AT-LSTM** (Wang et al. 2016b): The representation of a target is used to capture the associated words through standard attention which is employed on top of LSTM.
- **MemNet** (Tang, Qin, and Liu 2016): This model performs the standard attention for a certain number of times (multi-hops) before it catches the right sentiment.
- **RAM** (Chen et al. 2017): Similar to MemNet, it adds a recurrent function between multiple attentions to model the inner dependencies between different steps of attention.

Note that AT-LSTM, MemNet, RAM are all attention-based methods. The results of MemNet and RAM are retrieved from (Chen et al. 2017). The intuition behind multiple attentions is that there may be multiple semantic components that contribute to the sentiment of a particular target. Similarly, segmentation attention can also be viewed as "multiple attention" at word level.[2] With this setting, the model has the better capacity to model the dependencies between multiple attentions. Another point that distinguishes our model from them is that we use the binary indicator feature to encode the target's information at the input layer.

For the SemEval 2016 review datasets, our model may be the first approach that makes use of the attention mechanism. We compare it with two best models reported in the competition and two LSTM-based models.

- **XRCE** (Brun, Perez, and Roux 2016): This model involves many hand-crafted rules based on the syntactic features generated by a parser.
- **IIT-TUDA** (Kumar et al. 2016): It incorporates domain dependency graph features and a large amount of sentiment lexicons for each language.

Method	Laptop	Restaurant	Twitter
SVM	70.5	80.2	63.4
AdaRNN	-	-	66.3
AT-LSTM	68.9	77.2	-
MemNet	70.3	78.2	68.5
RAM	74.5	80.2	69.4
A-Softmax	68.8	76.9	66.0
SA-Softmax	69.0	77.1	66.2
SA-Softmax-P	69.1	77.8	66.5
A-LSTM	72.7	78.4	68.2
SA-LSTM	74.5	79.8	**69.9**
SA-LSTM-P	**75.1**	**81.6**	69.0

Table 2: Results on reviews from SemEval 2014 Task 4 and comments from Twitter in terms of accuracy (%).

- **LSTM, HP-LSTM** (Ruder, Ghaffari, and Breslin 2016): The BiLSTM is used to encode the sentence, then the first and last hidden states are combined for sentiment prediction. HP-LSTM adds a review-level LSTM to capture the information between sentences within the same review.

To see how well the BiLSTM and segmentation attention layer work, we implemented the basic versions that leave them out respectively. We also implemented the standard attention-based model for comparison.

- **A-Softmax**: Without using LSTM, this model directly performs attention mechanism on the input embeddings of each word. It can be viewed as the process that directly selects sentiment word from the sentence without considering the contextual information.
- **SA-Softmax, SA-Softmax-P**: This model replaces the standard attention of A-Softmax with segmentation attention. SA-Softmax-P adds penalty terms.
- **A-LSTM**: LSTM is used to capture the contextual information before the attention layer. This model is very similar to the AT-LSTM except that we used a different target representation.
- **SA-LSTM, SA-LSTM-P**: Segmentation attention layer is employed on top of the LSTM. This is the main model of this paper. Penalty terms are added to guide the learning process in SA-LSTM-P.

Training Details

We use the 300 dimension word embeddings from GloVe (Pennington, Socher, and Manning 2014) for English datasets. The pre-trained embeddings of other languages are

[1] Detailed statistics can be found in (Pontiki et al. 2014; 2016).

[2] Our model can also be extended to perform multiple segmentation attention at sentence level by stacking multiple such layers.

Method	Restaurant						Hotel
	English	Spanish	French	Turkish	Russia	Dutch	Arabic
XRCE	88.1	-	78.8	-	-	-	-
IIT-TUDA	86.7	83.6	72.2	84.3	73.6	77.0	81.7
LSTM	81.4	75.7	69.8	73.6	73.9	73.6	80.5
HP-LSTM	85.3	81.8	75.4	79.2	77.4	84.8	82.9
A-LSTM	86.5	86.5	81.8	**86.2**	81.3	85.6	86.5
SA-LSTM	88.1	83.8	81.9	78.6	81.1	86.1	86.7
SA-LSTM-P	**88.7**	**88.0**	**82.4**	83.7	**82.8**	**87.3**	**86.9**

Table 3: Results on review datasets from SemEval 2016 task 5 in terms of accuracy (%)

taken from (Ruder, Ghaffari, and Breslin 2016)[3] which are also used by the methods that we compare against. The dimension of target's binary indicator embedding is 30. We fixed the word embeddings in all the experiments since we found that it is very easy to get overfitting if we keep fine-tuning them. Dropout is also used after the input layer and it is tuned for each dataset. λ_1 is tuned between 0 and 1, λ_2 is chosen from [0, 0.2] with step size 0.04. For LSTM, we set the hidden dimension size to 300. One-sixth of training data is left out as the validation set for tuning hyper-parameters and doing model selection. The model is trained using stochastic gradient descent with the update rule of Adam (Kingma and Ba 2015).

Results

The main results based on the first group of datasets can be found in Table 2. Attention based models are effective without requiring hand-crafted features or an external parser compared with SVM and AdaRNN. Both the LSTM layer and the segmentation attention layer consistently improve the performance in all domains. Compared with MemNet and RAM, which both extends the standard attention with multi-hop, segmentation attention model gets a better performance since it can capture more structural information which is beneficial for this task. We can empirically conclude that in this task attending to sentiment spans at one time is a more natural and effective way than attending to one word at a time for multiple times. The penalty terms usually have positive effects on the segmentation attention layer except for the social comments from Twitter. We found that Twitter text is comparatively noisy and less-structured, and as a result, the introduction of regularizers tend to lead to the inclusion of wrong opinions. Also, from the results we can observe that the segmentation attention layer performs better when it is paired with an LSTM component.

Table 3 shows the results of review datasets across seven languages from SemEval 2016 Task 5. In line with the above findings, attention layer improves the performance of all datasets. Compared with the first two models which are both based on expensive manual feature engineering, the improvement is significant especially for low-resource languages where hand-crafted features are not adequate anymore. Our model is superior to conventional LSTM because of the effectiveness of attention mechanism. Furthermore, segmentation attention based model generally achieves better results compared with the standard attention-based model except for Turkish reviews[4], largely demonstrating the language insensitivity of our model. Also, segmentation attention consistently benefits from the additional regularizers and we can conclude that our assumption of the spans works across different languages.

Method	Laptop		Restaurant	
	Precision	Recall	Precision	Recall
A-Softmax	50.2	44.6	59.7	36.9
SA-Softmax	36.2	68.1	38.5	64.7
SA-Softmax-P	65.5	55.2	42.2	58.3
A-LSTM	48.4	47.9	56.5	53.7
SA-LSTM	25.5	75.4	21.2	78.3
SA-LSTM-P	49.1	58.7	39.4	59.9

Table 4: Performance (%) of unsupervised extraction of opinions at word level

Latent Opinions

We have observed the effectiveness of our proposed attention mechanism. However, does the attention layer focus on the true opinion expressions as desired? To understand this, we then take a further step to evaluate the latent opinions during predictions. We extract the sentiment words or spans, namely opinions that have a relatively significant effect on final prediction. For standard attention-based models, we extract a single opinion word by doing max operation on the attention layer. For segmentation attention, we extract the optimal opinion spans using Viterbi decoding algorithm which results in one or multiple opinion spans. Both are evaluated with the annotated opinions provided by (Wang et al. 2016a) at word level.

Main results are shown in Table 4. Compared with models that do not employ BiLSTM, we found that BiLSTM generally helps recall more opinion expressions since some sentiment words are context dependent. For example, "high" is positive when it is paired with "tech", but it becomes negative when paired with "price". BiLSTM gives the model the basic understanding of the sentence. Segmentation attention

[3] https://s3.amazonaws.com/aylien-main/data/multilingual-embeddings/index.html

[4] The size of Turkish dataset is relatively small, the difference between A-LSTM and SA-LSTM-P is only 4 instances.

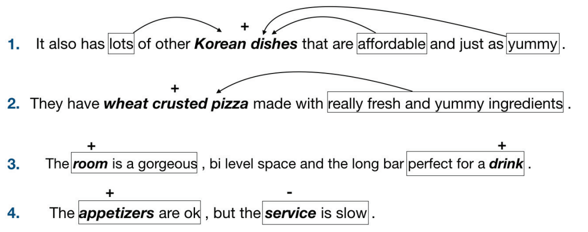
1. It also has lots of other Korean dishes that are affordable and just as yummy .
+
2. They have wheat crusted pizza made with really fresh and yummy ingredients .
+
3. The room is a gorgeous , bi level space and the long bar perfect for a drink .
+
+
4. The appetizers are ok , but the service is slow .
+
-

Figure 3: Visualization of the extracted opinions from segmentation attention model. Targets are in bold and extracted opinions are highlighted with boxes, “+” and “-” on the target indicate positive and negative sentiment respectively.

works in a similar way by modeling the dependencies between opinion words directly. Since these datasets are relatively small, BiLSTM may be limited in capturing contextual information. The structural bias that segmentation attention brings to the model helps it identify more opinion expressions as a result. Furthermore, segmentation attention usually performs better when coupled with BiLSTM since it is easier to model the interactions if contextual information is encoded in each representation.

Compared with standard attention, segmentation attention can naturally recall more opinion expressions since the model captures a sequence of words rather than individual words. Intuitively, more identified sentiment words or spans can reveal more sentiment information for the model to make the right decision. However, high recall also results in low precision when segmentation attention is plugged in. Two regularizers are introduced to balance them. They help the model focus on correct opinions based on the assumption that opinions should be short and coherent spans.

Analyses

Case Studies

Figure 3 gives some examples of extracted opinions through using our proposed segmentation attention. We found that the model is able to impressively attend to the correct opinion expressions in various cases.

The internal segmentation attention layer is expected to select the opinions based on both content and structure information. As we can see from the first example, the segmentation attention successfully detects multiple sentiment words towards it. The second example shows that the model can successfully attend to a coherent span when the opinion consists of a consecutive sequence of words in the sentence.

In the third example, two targets hold the same positive sentiment. In this case, the correct prediction will always be made regardless of which sentiment words the model attends to. Nevertheless, our model still accurately attends to the correct opinion span that contains both the target and its opinion terms. Similarly, as we can see from the fourth example, the model is also able to handle cases where targets within the same sentence are associated with different sentiments. From these examples, we can observe that the proposed model is capable of capturing shallow structural information so as to perform sentiment classification.

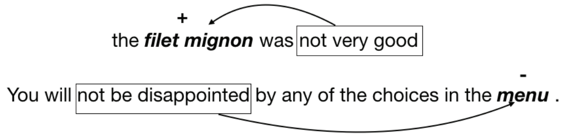
+
the filet mignon was not very good
-
You will not be disappointed by any of the choices in the menu .

Figure 4: Examples with negation words and the model fails to detect the sentiment of the span.

Error Analysis

The internal attention layer of our model also makes it convenient to trace back the errors that the model makes. Based on our analysis, we found that errors can be broadly categorized into two types: attention error and representation error.

The first type of errors come from the reason that the model fails to attend to the right span because of various reasons. One of them is that the model tends to assign sentiment to words that express intensity, such as “really” and “sure”, since they are usually parts of sentiment expressions. It would be interesting to see if incorporating syntactic features can improve the performance since syntactic information would hopefully further guide the learning process of latent opinions.

Most of the remaining errors belong to the latter category – representation errors. Though the model can attend to the correct opinion spans, it does not necessarily mean it could always figure out the correct sentiment the opinion expresses. For example in Figure 4, the model successfully identifies the opinion span but it fails to capture the correct sentiment that involves negation words. Recall in Equation 16 and 17, opinion vectors are calculated based on simple sum operations over word representations. Essentially, the model relies on the LSTM to capture contextual information when generating the word representation. However, it appears that certain stronger mechanism needs to be devised in the future in order to capture negation scope better.

Conclusion

In this work, we propose a novel model that learns the latent opinions based on segmentation attention layer for aspect-level sentiment classification. The internal selection layer not only benefits the classification but also gives the great interpretation of how the model works. Experiments on extensive datasets show that the model consistently achieves the state-of-art performance. We further evaluate the extracted latent opinions by comparing them with annotated opinions to show how segmentation attention layer affects the model.

Future work includes applying the proposed attention model to other tasks such as question answering. Specifically, to predict or generate the answer for a particular question, the model can use the proposed segmentation attention mechanism to search for latent expression spans related to the question in an unsupervised manner – a step that is analogous to the process of answering questions by humans.

Acknowledgements

We would like to thank all reviewers for their thoughtful comments. This work was done when the first author was visiting Singapore University of Technology and Design. This work is supported by MOE Tier 1 grant SUTDT12015008.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Brun, C.; Perez, J.; and Roux, C. 2016. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis.

Chen, P.; Xu, B.; Yang, M.; and Li, S. 2016. Clause sentiment identification based on convolutional neural network with context embedding. In *ICNC-FSKD*.

Chen, P.; Sun, Z.; Bing, L.; and Wei, Y. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP*.

Ding, X.; Liu, B.; and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *WSDM*.

Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; and Xu, K. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL*.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*.

He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent twitter sentiment classification. In *ACL*.

Kim, Y.; Denton, C.; Hoang, L.; and Rush, A. M. 2017. Structured attention networks. In *ICLR*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews.

Kumar, A.; Kohail, S.; Kumar, A.; Ekbal, A.; and Biemann, C. 2016. Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. In *EMNLP*.

Li, H., and Lu, W. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *AAAI*.

Li, F.; Han, C.; Huang, M.; Zhu, X.; Xia, Y.-J.; Zhang, S.; and Yu, H. 2010. Structure-aware review mining and summarization. In *ACL*.

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

Mitchell, M.; Aguilar, J.; Wilson, T.; and Van Durme, B. 2013. Open domain targeted sentiment.

Pang, B.; Lee, L.; et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval 2014*, 27–35.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *SemEval-2016*, 19–30.

Ruder, S.; Ghaffari, P.; and Breslin, J. G. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *EMNLP*.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; Potts, C.; et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *NIPS*.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Target-dependent sentiment classification with long short term memory. In *COLING*.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *EMNLP*.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *AAAI*.

Wang, Y.; Huang, M.; zhu, x.; and Zhao, L. 2016b. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Yu, A. W.; Lee, H.; and Le, Q. V. 2017. Learning to skim text. In *ACL*.

Zhao, W. X.; Jiang, J.; Yan, H.; and Li, X. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *EMNLP*.