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

Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM

Yukun Ma,* **Haiyun Peng,†** **Erik Cambria†**
*Rolls-Royce@NTU Corporate Lab, Nanyang Technological University
†School of Computer Science and Engineering, Nanyang Technological University

Abstract

Analyzing people's opinions and sentiments towards certain aspects is an important task of natural language understanding. In this paper, we propose a novel solution to targeted aspect-based sentiment analysis, which tackles the challenges of both aspect-based sentiment analysis and targeted sentiment analysis by exploiting commonsense knowledge. We augment the long short-term memory (LSTM) network with a hierarchical attention mechanism consisting of a target-level attention and a sentence-level attention. Commonsense knowledge of sentiment-related concepts is incorporated into the end-to-end training of a deep neural network for sentiment classification. In order to tightly integrate the commonsense knowledge into the recurrent encoder, we propose an extension of LSTM, termed Sentic LSTM. We conduct experiments on two publicly released datasets, which show that the combination of the proposed attention architecture and Sentic LSTM can outperform state-of-the-art methods in targeted aspect sentiment tasks.

Introduction

In recent years, sentiment analysis (Cambria et al. 2017a) has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media. Sentiment analysis is a branch of affective computing research (Poria et al. 2017) that aims to classify text into either positive or negative, but sometimes also neutral (Chaturvedi et al. 2017). Most of the literature is on English language but recently an increasing number of publications is tackling the multilinguality issue (Lo et al. 2017).

While most works approach it as a simple categorization problem, sentiment analysis is actually a suitcase research problem (Cambria et al. 2017b) that requires tackling many natural language processing (NLP) tasks, including named entity recognition (Ma, Cambria, and Gao 2016), word polarity disambiguation (Xia et al. 2015), personality recognition (Majumder et al. 2017), sarcasm detection (Poria et al. 2016), and aspect extraction. The last one, in particular, is an extremely important subtask that, if ignored, can consistently reduce the accuracy of sentiment classification in the presence of multiple opinion targets.

Hence, aspect-based sentiment analysis (ABSA) (Pontiki et al. 2014; 2016; Poria, Cambria, and Gelbukh 2016) extends the typical setting of sentiment analysis with a more realistic assumption that polarity is associated with specific aspects (or product features) rather than the whole text unit. For example, in the sentence "The design of the space is good but the service is horrible", the sentiment expressed towards the two aspects ("space" and "service") is completely opposite. Through aggregating sentiment analysis with aspects, ABSA allows the model to produce a fine-grained understanding of people's opinion towards a particular product.

Targeted (or target-dependent) sentiment classification (Tang et al. 2016; Dong et al. 2014; Wang et al. 2017), instead, resolves the sentiment polarity of a given target in its context, assuming that a sentence might express different opinions towards different targeted entities. For instance, in the sentence "I just log on my [facebook]. [Transformers] is boring", the sentiment expressed towards [Transformers] is negative, while there is no clear sentiment for [facebook]. Recently, targeted ABSA (Saeidi et al. 2016) has attempted to tackle the challenges of both ABSA and targeted sentiment analysis. The task is to jointly detect the aspect category and resolve the polarity of aspects with respect to a given target.

Deep learning methods (Nguyen and Shirai 2015; Wang et al. 2016; Tang et al. 2016; Tang, Qin, and Liu 2016; Wang et al. 2017) have achieved great accuracy when applied to ABSA and targeted sentiment analysis. Especially, neural sequential models, such as long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997), are of growing interest for their capacity of representing sequential information. Moreover, most of these sequence-based methods incorporate the attention mechanism, which has its root in the alignment model of machine translation (Bahdanau, Cho, and Bengio 2014). Such mechanism takes an external memory and representations of a sequence as input and produces a probability distribution quantifying the concerns in each position of the sequence.

Despite these advances in sentiment analysis, we identify three problems remaining unsolved in current state-of-the-art methods. Firstly, a given target might consist of multiple instances (mentions of the same target) or multiple words in a sentence, existing research assumes all instances are of equal importance and simply computes an average vector

over such instances. This oversimplification conflicts with the fact that one or more instances of the target are often more tightly tied with sentiment than others. Secondly, hierarchical attention exploited by existing methods only implicitly models the process of inferring the sentiment-bearing words related to the given target and aspect as black-box. Last but not least, existing research falls short in effectively incorporating into the deep neural network external knowledge, e.g., affective or commonsense knowledge, that could directly contribute to the identification of aspects and sentiment polarity. Without any constraints, moreover, the global attention model might tend to encode task-irrelevant information. To address these problems, our method simultaneously learns a target-specific instance attention as well as a global attention. In particular, our contribution is three-fold:

1. We propose a hierarchical attention model that explicitly attends to first the targets and then the whole sentence;
2. We extend the classic LSTM cell with components accounting for integration with external knowledge;
3. We incorporate affective commonsense knowledge into a deep neural network.

Related Work

In this section, we survey multiple research areas related to the proposed framework, namely: ABSA, targeted sentiment analysis, targeted ABSA, and finally works on incorporating external knowledge into deep neural models.

Aspect-Based Sentiment Analysis

ABSA is the task of classifying sentiment polarity with respect to a set of aspects. The biggest challenge faced by ABSA is how to effectively represent the aspect-specific sentiment information of the whole sentence. Early works on ABSA have mainly relied on feature-engineering to characterize sentences (Wagner et al. 2014; Kiritchenko et al. 2014). Motivated by the success of deep learning in representation learning, many recent works (Dong et al. 2014; Lakkaraju, Socher, and Manning 2014; Nguyen and Shirai 2015; Wang et al. 2016) utilize deep neural networks to generate sentence embeddings (dense vector representation of sentences) which are then fed to a classifier as a low-dimensional feature vector. Moreover, the representation can be enhanced by using the attention mechanism (Wang et al. 2016), which is typically a multi-layer neural network taking as input the word sequence and aspects. For each word of the sentence, the attention vector quantifies its sentiment salience as well as the relevance to the given aspect. The resulting sentiment representation benefits from the attention mechanism for it overcomes the shortcoming of recurrent neural networks (RNNs), which suffer from information loss when only one single output (e.g., the output at the end of the sequence) is used by the classifier.

Targeted Sentiment Analysis

Targeted sentiment analysis aims to analyze sentiment with respect to targeted entities in the sentence. It is thus critical for targeted sentiment analysis methods, e.g., the target-dependent LSTM (TDLSTM) and target connection LSTM (TCLSTM) (Tang et al. 2016), to model the interaction between sentiment targets and the whole sentence. In order to obtain the target-dependent sentence representation, TDLSTM directly uses the hidden outputs of a bidirectional-LSTM sentence encoders panning the target mentions, while TCLSTM extends TDLSTM by concatenating each input word vector with a target vector. Similar to ABSA, attention models are also applicable to targeted sentiment analysis. Rather than using a single level of attention, deep memory networks (Tang, Qin, and Liu 2016) and recurrent attention models (Chen et al. 2017) have achieved superior performance by learning a deep attention over the single-level attention, as multiple passes (or hops) over the input sequence could refine the attended words again and again to find the most important words. All existing approaches have either ignored the problem of multiple target instances (or words) or simply used an averaging vector over target expressions (Tang, Qin, and Liu 2016; Wang et al. 2017). Unlike such approaches, our method weights each target word with an attention weight so that a given target is represented by its most informative components.

Targeted Aspect-Based Sentiment Analysis

Two baseline systems (Saeidi et al. 2016) are proposed together with SentiHood: a feature-based logistic regression model and a LSTM-based model. The feature-based logistic regression model uses feature templates including n-grams tokens and POS tags extracted from the context of instances. The LSTM baseline can be seen as an adaptation of TDLSTM that simply uses the hidden outputs at the position of target instances assuming that all target instances are equally important.

Incorporating External Knowledge

External knowledge base has been typically used as a source of features (Ratinov and Roth 2009; Rahman and Ng 2011; Nakashole and Mitchell 2015). Most recently, neural sequential models (Ahn et al. 2016; Yang and Mitchell 2017) leverage the lower-dimensional continuous representation of knowledge concepts as additional inputs. However, these approaches have treated the computation of neural sequential models as a black-box without tight integration of knowledge and computational structure. The proposed model, termed Sentic LSTM, is inspired by (Xu et al. 2016), which adds a knowledge recall gate to the cell state of LSTM. However, our method differs from (Xu et al. 2016) in the way of using external knowledge to generate the hidden outputs and controlling the information flow.

Methodology

In this section, we describe the proposed attention-based neural architecture in detail: we first proposed the task definition of targeted ABSA, followed by an overview of the whole neural architecture; afterwards, we describe instance attention and global attention model; lastly, we describe the proposed knowledge-embedded extension of LSTM cell.

Task Definition

A sentence s consists of a sequence of words. Similar to (Wang et al. 2017), we consider all mentions of the same target as a single target. A target t composed of m words in sentence s, denoted as $T = \{t_1, t_2, \cdots, t_i, \cdots, t_m\}$ with t_i referring to the position of ith word in the target expression, the task of targeted ABSA can be divided into two subtasks. Firstly, it resolves the aspect categories of t belonging to a predefined set. Secondly, it classifies the sentiment polarity with respect to each aspect category associated with t.

For example, the sentence "I live in [West London] for years. I like it and it is safe to live in much of [west London]. Except [Brent] maybe. " contains two targets, $[WestLondon]$ and $[Brent]$. Our objective is to detect the aspects and classify the sentiment polarity. The desired output for $[WestLondon]$ is ['*general*':**positive**; '*safety*':**positive**], while output for $[Brent]$ should be ['*general*':**negative**; '*safety*':**negative**].

Overview

In this section, we provide an overview of the proposed method. Our neural architecture consists of two components: the sequence encoder and a hierarchical attention component.

Fig. 1 illustrates how the neural architecture works. Given a sentence $s = \{w_1, w_2, \cdots, w_L\}$, a look-up operation is first performed to convert input words into word embeddings $\{v_{w_1}, v_{w_2}, \cdots, v_{w_L}\}$. The sequence encoder, which is based on a bidirectional LSTM, transforms the word embeddings into a sequence of hidden outputs. The attention component is built on top of the hidden outputs. The target-level attention takes as input the hidden outputs at the positions of target expression (highlighted in brown) and computes a self-attention vector over these words.

The output of target-level attention component is a representation of the target. Afterwards, the target representation together with the aspect embeddings is used for computing a sentence-level attention transforming the whole sentence into a vector. The sentence-level attention component returns one sentence vector for each aspect and target pair. The aspect-based sentence vector is then fed into the corresponding multi-class (e.g., None, Neural, Negative, and Positive for a 4-class setting; or None, Negative, and Positive for a 3-class setting) classifier to resolve the sentiment polarity.

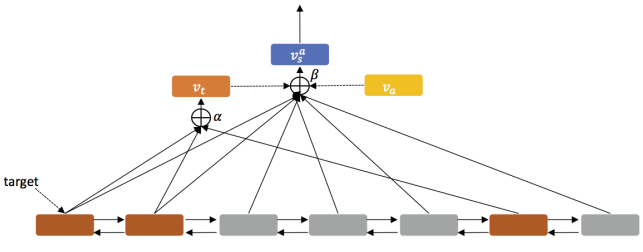

Figure 1: Overview of the attentive neural architecture

Long Short-Term Memory Network

The sentence is encoded using an extension of RNN (Schuster and Paliwal 1997), termed LSTM (Hochreiter and Schmidhuber 1997), which was firstly introduced by (Hochreiter and Schmidhuber 1997) to solve the vanishing and exploding gradient problem faced by the vanilla RNN. A typical LSTM cell contains three gates: forget gate, input gate and output gate. These gates determine the information to flow in and flow out at the current time step. The mathematical representations of the cell are as follows:

$$\begin{aligned} f_i &= \sigma(W_f[x_i, h_{i-1}] + b_f) \\ I_i &= \sigma(W_I[x_i, h_{i-1}] + b_I) \\ \widetilde{C}_i &= tanh(W_C[x_i, h_{i-1}] + b_C) \\ C_i &= f_i * C_{i-1} + I_i * \widetilde{C}_i \\ o_i &= \sigma(W_o[x_i, h_{i-1}] + b_o) \\ h_i &= o_i * tanh(C_i) \end{aligned} \tag{1}$$

where f_i, I_i and o_i are the forget gate, input gate and output gate, respectively. W_f, W_I, W_o, b_f, b_I and b_o are the weight matrix and bias scalar for each gate. C_i is the cell state and h_i is the hidden output. A single LSTM typically encodes the sequence from only one direction. However, two LSTMs can also be stacked to be used as a bidirectional encoder, referred to as bidirectional LSTM. For a sentence $s = \{w_1, w_2, \cdots, w_L\}$, bidirectional LSTM produces a sequence of hidden outputs,

$$H = [h_1, h_2 ... h_L] = \begin{bmatrix} \overrightarrow{h}_1 & \overrightarrow{h}_2 & \cdots & \overrightarrow{h}_L \\ \overleftarrow{h}_1 & \overleftarrow{h}_2 & \cdots & \overleftarrow{h}_L \end{bmatrix}$$

where each element of H is a concatenation of the corresponding hidden outputs of both forward and backward LSTM cells.

Target-Level Attention

Based on the attention mechanism, we calculate an attention vector for a target expression. A target might consist of a consecutive or non-consecutive sequence of words, denoted as $T = \{t_1, t_2, \cdots, t_m\}$, where t_i is the location of an individual word in a target expression. The hidden outputs corresponding to T is denoted as $H' = \{h_{t_1}, h_{t_2}, \cdots, h_{t_m}\}$. We compute the vector representation of a target t as

$$v_t = H'\alpha = \sum_j \alpha_j h_{t_j} \tag{2}$$

where the target attention vector $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_m\}$ is distributed over target word sequence T. The attention vector α is a self-attention vector that takes nothing but the hidden output itself as input. The attention vector α of target expression is computed by feeding the hidden output into a bi-layer perceptron, as shown in Equation 3.

$$\alpha = softmax(W_a^{(2)} tanh(W_a^{(1)} H')) \tag{3}$$

where $W_a^{(1)} \in R^{d_m \times d_h}$ and $W_a^{(2)} \in R^{1 \times d_m}$ are parameters of the attention component.

Sentence-Level Attention Model

Following the target-level attention, our model learns a target-and-aspect-specific sentence attention over all the words of a sentence. Given a sentence s of length L, the hidden outputs are denoted as $H = [h_1, h_2, \cdots, h_L]$. An attention model computes a linear combination of the hidden vectors into a single vector, i.e.,

$$v^a_{s,t} = H\beta = \sum_i \beta_i h_i \tag{4}$$

where the vector $\beta = [\beta_1, \beta_2, \cdots, \beta_L]$ is called the sentence-level attention vector. Each element β_i encodes the salience of the word w_i in the sentence s with respect to the aspect a and target T. Existing research on targeted sentiment analysis or ABSA mostly uses targets or aspect terms as queries.

At first, each h_i is transformed to a d_m dimensional vector by a multi-layer neural network with a $tanh$ activation function, followed by a dense softmax layer to generate a probability distribution over the words in sentence s, i.e.,

$$\beta_a = softmax(v_a^T tanh(W_m(H' \odot v_t))) \tag{5}$$

where v_a is the aspect embedding of aspect a, $H \odot v_t$ is the operation concatenating v_t to each h_i; $W_m^{(1)} \in R^{d_m \times d_h}$ is the matrix mapping row vectors of H to a d_m dimensional space, and $W_m^{(2)} \in R^{1 \times d_m}$ maps each new row vector to a unnormalized attention weight.

Commonsense Knowledge

In order to improve the accuracy of sentiment classification, we use commonsense knowledge as our knowledge source to be embedded into the sequence encoder. In particular, we use SenticNet (Cambria et al. 2016), a commonsense knowledge base that contains 50,000 concepts associated with a rich set of affective properties (Table 1). These affective properties provide not only concept-level representation but also semantic links to the aspects and their sentiment. For example, the concept 'rotten fish' has property "KindOf-food" that directly relates with aspects such as 'restaurant' or 'food quality', but also emotions, e.g., 'joy', which can support polarity detection (Fig. 2).

However, the high dimensionality of SenticNet hinders it from being used in deep neural models. AffectiveSpace (Cambria et al. 2015) has been built to map the concepts of SenticNet to continuous low-dimensional embeddings without losing the semantic and affective relatedness of the original space. Based on this new space of concepts, we embed concept-level information into deep neural sequential models to better classify both aspects and sentiment in natural language text.

Table 1: Example of SenticNet assertions

SenticNet	IsA-pet	KindOf-food	Arises-joy	...
dog	0.981	0	0.789	...
cupcake	0	0.922	0.910	...
rotten fish	0	0.459	0	...
police man	0	0	0	...
win lottery	0	0	0.991	...

Sentic LSTM

In order to leverage SenticNet's affective commonsense knowledge efficiently, we propose an affective extension of LSTM, termed Sentic LSTM. It is reasonable to assume that SenticNet concepts contain information complementary to the textual word sequence as, by definition, commonsense knowledge is about concepts that are usually taken for granted and, hence, absent from text. Sentic LSTM aims to entitle the concepts with two important roles: 1) assisting with the filtering of information flowing from one time step to the next and 2) providing complementary information to the memory cell. At each time step i, we assume that a set of knowledge concept candidates can be triggered and mapped to a d_c dimensional space. We denote the set of K concepts as $\{\mu_{i,1}, \mu_{i,2}, \cdots, \mu_{i,K}\}$. First, we combine the candidate embeddings into a single vector as follows:

$$\mu_i = \frac{1}{K} \sum_j \mu_{i,j} \tag{6}$$

As we realized that there are only up to 4 extracted concepts for each time step, we simply use the average vector (although a more sophisticated attention model can also be easily employed to replace the averaging function).

$$\begin{aligned}
f_i &= \sigma(W_f[x_i, h_{i-1}, \mu_i] + b_f) \\
I_i &= \sigma(W_I[x_i, h_{i-1}, \mu_i] + b_I) \\
\widetilde{C}_i &= tanh(W_C[x_i, h_{i-1}] + b_C) \\
C_i &= f_i * C_{i-1} + I_i * \widetilde{C}_i \\
o_i &= \sigma(W_o[x_i, h_{i-1}, \mu_i] + b_o) \\
o_i^c &= \sigma(W_{co}[x_i, h_{i-1}, \mu_i] + b_{co}) \\
h_i &= o_i * tanh(C_i) + o_i^c * tanh(W_c \mu_i)
\end{aligned} \tag{7}$$

Our affective extension of LSTM is illustrated in Equation 7. At first, we assume that affective concepts are meaningful cues to control the information of token-level information. For example, a multi-word concept 'rotten fish' might indicate that the word 'rotten' is a sentiment-related modifier of its next word 'fish' and, hence, less information should be filtered out at next time step. We thus add knowledge concepts to the forget, input, and output gate of standard LSTM to help filtering the information. The presence of affective concepts in the input gate is expected to prevent the memory cell from being affected by input tokens conflicting with pre-existing knowledge. Similarly, the output gate uses such knowledge to filter out irrelevant information stored in the memory.

Another important feature of Sentic LSTM is based on the assumption that the information from the concept-level output is complementary to the token level. Therefore, we extended the regular LSTM with an additional knowledge output gate o_i^c to output concept-level knowledge complementary to the token-level memory. Since AffectiveSpace is learned independently, we leverage a transformation matrix $W_c \in R^{d_h \times d_\mu}$ to map it to the same space as the memory outputs. In other words, o_i^c models the relative contributions of token level and concept level.

Moreover, we notice that $o_i^c * tanh(W_c\mu_i)$ actually resembles the functionality of the sentinel vector used by (Yang and Mitchell 2017), which allows the model to choose whether to use affective knowledge or not.

Prediction and Parameter Learning

The objective to train our classier is defined as minimizing the sum of the cross-entropy losses of prediction on each target-aspect pair, i.e.,

$$\mathcal{L}_s = \frac{1}{|D|} \sum_{s \in D} \sum_{t \in s} \sum_{a \in A} \log p_{c,t}^a$$

where A is the set of predefined aspects, and $p_{c,t}^a$ is the probability of the gold-standard polarity class c given target t with respect to a sentiment category a, which is defined by a softmax function,

$$p_{c,t}^a = softmax(W^p v_{s,t}^a + b_s^a)$$

where W^p and b_s^a are the parameters to map the vector representation of target t to the polarity label of aspect a. To avoid overfitting, we add a dropout layer with dropout probability of 0.5 after the embedding layer. We stop the training process of our model after 10 epochs and select the model that achieves the best performance on the development set.

Experiments

Dataset and Resources

We evaluate our method on two datasets: SentiHood (Saeidi et al. 2016) and a subset of Semeval 2015 (Pontiki et al. 2015). SentiHood was built by querying Yahoo! Answers with location names of London city. Table 2 shows statistics of SentiHood. The whole dataset is split into train, test, and development set by the authors. Overall, the entire dataset contains 5,215 sentences, with 3,862 sentences containing a single target and 1,353 sentences containing multiple targets. It also shows that there are approximately two third of targets annotated with aspect-based sentiment polarity (train set: 2476 out of 2977; test set:1241 out of 1898; development set: 619 out of 955). On average, each sentiment-bearing target has been annotated with 1.37 aspects. To show the generalizability of our methods, we build a subset of the dataset used by Semeval-2015. We remove sentences containing no targets as well as $NULL$ targets. To be comparable with SentiHood, we combine targets with the same surface form within the same sentence as mentions of the same target. In total, we have 1,197 targets left in the training set and 542 targets left in the testing set. On average, each target has 1.06 aspects.

Table 2: SentiHood dataset

	Train	Dev	Test
Targets	3,806	955	1,898
Targets w/ Sentiment	2,476	619	1,241
Aspects per Target(w/ Sentiment)	1.37	1.37	1.35

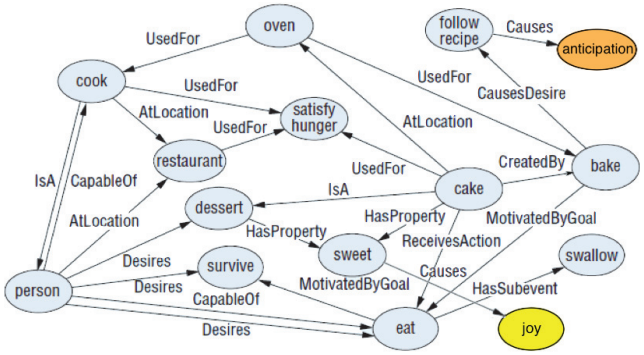

Figure 2: A sketch of SenticNet semantic network

To inject the commonsense knowledge, we use a syntax-based concept parser[1] to extract a set of concept candidates at each time step, and use AffectiveSpace[2] as the concept embeddings. In case no concepts are extracted, a zero vector is used as the concept input.

Experiment Setting

We evaluate our method on two sub-tasks of targeted ABSA: 1) aspect categorization and 2) aspect-based sentiment classification. Following Saeidi et al. (Saeidi et al. 2016), we treat the outputs of aspect-based classification as hierarchical classes. For aspect categorization, we output the label (e.g., in the 3-class setting, it outputs 'Positive', 'Negative', or 'None') with the highest probability for each aspect. For aspect-based sentiment classification, we ignore the scores of 'None'. For evaluating the aspect-based sentiment classification, we simply calculate the accuracy averaged over aspects. We evaluate aspect categorization as a multi-label classification problem so that results are averaged over targets instead of aspects.

We evaluate our methods and baseline systems using both loose and strict metrics. We report scores of three widely used evaluation metrics of multi-label classifier: Macro-F1, Micro-F1, and strict Accuracy. Given the dataset D, the ground-truth aspect categories of the target $t \in D$ is denoted as Y_t, while the predicted aspect categories denoted as $\widehat{Y}_t$. The three metrics can be computed as

- Strict accuracy (Strict Acc.): $\frac{1}{D} \sum_{t \in D} \sigma(Y_t = \widehat{Y}_t)$, where $\sigma(\cdot)$ is an indicator function.
- Macro-F1 = $2\frac{\text{Ma-P} \times \text{Ma-R}}{\text{Ma-P} + \text{Ma-R}}$, which is based on Macro-Precision (Ma-P) and Micro-Recall (Ma-R) with Ma-P = $\frac{1}{|D|} \sum_{t \in D} \frac{|Y_t \cap \widehat{Y}_t|}{\widehat{Y}_t}$, and Ma-R= $\frac{1}{|D|} \sum_{t \in D} \frac{|Y_t \cap \widehat{Y}_t|}{Y_t}$.
- Micro-F1 = $2\frac{\text{Mi-P} \times \text{Mi-R}}{\text{Mi-P} + \text{Mi-R}}$, which is based on Micro-Precision (Mi-P) and Micro-Recall (Mi-R), where Mi-P= $\frac{\sum_{t \in D} |Y_t \cap \widehat{Y}_t|}{\sum_{t \in D} \widehat{Y}_t}$, and Mi-R= $\frac{\sum_{t \in D} |Y_t \cap \widehat{Y}_t|}{\sum_{t \in D} Y_t}$.

[1] http://github.com/senticnet
[2] http://sentic.net/downloads

Table 3: System performance on SentiHood dataset

	Aspect Categorization						Sentiment	
	Strict Acc. (%)		Macro F1 (%)		Micro F1 (%)		Sentiment Acc. (%)	
	dev	test	dev	test	dev	test	dev	test
TDLSTM	50.27	50.83	59.03	58.17	55.72	55.78	82.60	81.82
LSTM + TA	54.17	52.02	62.90	61.07	60.56	59.02	83.80	84.29
LSTM + TA + SA	68.83	66.42	79.36	76.69	79.14	76.64	86.00	86.75
LSTM + TA + DMN SA	60.66	60.14	68.89	70.19	67.28	68.37	84.80	83.36
LSTM + TA + SA + KB Feat	**69.38**	64.76	**80.00**	76.33	**79.79**	76.08	87.00	88.70
LSTM + TA + SA + KBA	68.08	65.12	78.68	76.40	78.73	76.46	87.40	87.98
Recall LSTM + TA + SA	68.64	64.66	78.44	75.61	78.53	75.91	86.80	86.85
Sentic LSTM + TA + SA	69.20	**67.43**	78.84	**78.18**	79.09	**77.66**	**88.80**	**89.32**

Performance Comparison

We compare our proposed method with the methods that have been proposed for targeted ABSA as well as methods proposed for ABSA or targeted sentiment analysis but applicable to targeted ABSA.

Furthermore, we also compare the performances of several variants of our proposed method in order to highlight our technical contribution. We run the model for multiple times and report the results that perform best in the development set. For Semeval-2015 dataset, we report the results of the final epoch.

- **TDLSTM**: TDLSTM (Tang et al. 2016) adopts Bi-LSTM to encode the sequential structure of a sentence and represents a given target using a vector averaged on the hidden outputs of target instances.
- **LSTM + TA**: Our method learns an instance attention on top of the outputs of LSTM to model the contribution of each instance.
- **LSTM + TA + SA**: In addition to target instance attention, we add a sentence-level attention to the model.
- **LSTM + TA + DMN SA**: The sentence-level attention is replaced by a dynamic memory network with multiple hops (Tang, Qin, and Liu 2016). We run the memory network with different numbers of hops and report the result with 4 hops (best performance on development set of SentiHood). We exclude the case of zero hops as it corresponds to Bi-LSTM + TA + SA.
- **LSTM + TA + SA + KB Feat**: Concepts are fed into the input layer as additional features.
- **LSTM + TA + SA + KBA**: This is an integration of the method proposed by (Yang and Mitchell 2017), which learns an attention over the concept embeddings (the embeddings are combined with the hidden output before being fed into the classifier).
- **Recall LSTM + TA + SA**: LSTM is extended with a recall knowledge gate as in (Xu et al. 2016).
- **Sentic LSTM + TA + SA**: The encoder is replaced with the proposed knowledge-embedded LSTM.

The word embedding of the input layer is initialized by a pre-trained skip-gram model (Mikolov et al. 2013) with 150 hidden units on a combination of Yelp[3] and Amazon review dataset (He and McAuley 2016) and 50 hidden units for the bi-directional LSTM.

Results of Attention Model

Table 3 and Table 4 show the performance on SentiHood and Semeval-15 dataset, respectively. In comparison with the non-attention baseline (Bi-LSTM+Avg.), we can find that our best attention-based model significantly improves aspect categorization (by more than 20%) and sentiment classification (approximately 10%) on SentiHood. However, it is notable that, on the Semeval-2015 dataset, the improvement is relatively smaller. We conjecture the reason is that SentiHood has masked the target as a special word "LOCATION", which resulted less informative than the full name of aspect targets that are used by Semeval-2015.

Hence, using only the hidden outputs regarding the target does not suffice to represent the sentiment of the whole sentence in SentiHood. Compared with target averaging model, the target-level attention achieves some improvement (even though not significant), as the target attention is capable of identifying the part of target expressions with higher sentiment salience. On the other hand, it is notable that the two-step attention achieves significant improvement on both aspect categorization and sentiment classification, indicating that the target- and aspect-dependent sentence attention could retrieve information relevant to both tasks.

To our surprise, using multiple hops in the sentence-level attention fails to bring in any improvement. The performance even falls down significantly on Semeval-2015 with a much smaller number of training instances but larger aspect set than SentiHood. We conjecture the reason is that using multi-hops increases the number of parameter to learn, which makes it less applicable to small and sparse datasets.

Visualization of Attention

We visualize the attention vectors of sentence-level attention in Figure 3 with respect to "Transition-location" and "Price" aspects. The two attention vectors have encoded quite different concerns in the word sequence.

[3] http://yelp.com.sg/dataset/challenge

Table 4: System performance on Semeval-2015 dataset

	Aspect Categorization			Sentiment
	Strict Acc.	Macro F1	Micro F1	Sentiment Acc.
TDLSTM	65.49	70.56	69.00	68.57
LSTM+TA	66.42	71.71	70.06	69.24
LSTM+TA+SA	63.46	70.73	66.18	74.28
LSTM+TA+DMN SA	48.33	52.73	51.39	69.07
LSTM+TA+SA+KB Feat	65.68	74.46	70.71	76.13
LSTM+TA+SA+KBA	**67.34**	74.36	71.78	73.10
Recall LSTM + TA + SA	66.05	72.90	69.66	74.11
Sentic LSTM + TA + SA	67.34	**76.44**	**73.82**	**76.47**

In the first example, the 'Transition-location' attention attends to the word "long", which is expressing a negative sentiment towards the target. In comparison, the 'Price' attention attends more to the word 'cheap', which is related to the aspect. That is to say, the two attention vectors are capable of distinguishing information related to different aspects. As visualized in Figure 4, the target-level attention is capable of selecting the part of target expression of which the aspect or sentiment is easier to be resolved.

Results of Knowledge-Embedded LSTM

It can be seen from Table 3 and 4 that injecting knowledge into the model improves the performance in general. Since AffectiveSpace encodes the information about affective properties that are semantically related to the aspects, it is reasonable to find out that it can improve performance on both tasks. The results show that our proposed Sentic LSTM outperforms baseline methods, even if not significantly.

An important outcome of the experiments is that Sentic LSTM significantly outperforms a baseline (LSTM + TA + SA + KB feat) feeding the knowledge features to the input layer, which confirms the efficacy of using a knowledge output gate to control the flow of background knowledge. Furthermore, the superior performance of Sentic LSTM over Recall LSTM and KBA indicates that the activated knowledge concepts can also help filtering the information that conflicts with the background knowledge.

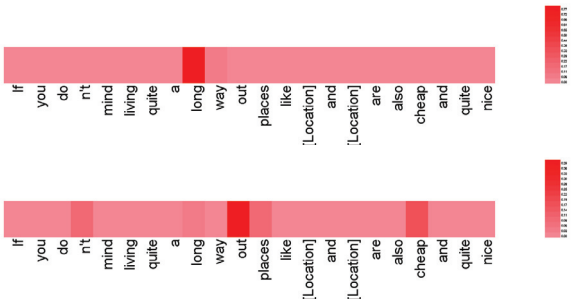

Figure 3: Example of sentence-level attention

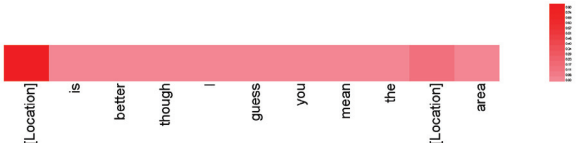

Figure 4: Example of target-level attention

Conclusion

In this paper, we proposed a neural architecture for the task of targeted ABSA. We explicitly modeled the attention as a two-step model which encodes targets and full sentence. The target-level attention learns to attend to the sentiment-salient part of a target expression and generates a more accurate representation of the target, while the sentence-level attention searches for the target- and aspect-dependent evidence over the full sentence. Moreover, we proposed an extension of the LSTM cell so that it could more effectively incorporate affective commonsense knowledge when encoding the sequence into a vector. In the future, we would like to collectively analyze the sentiment of multiple targets co-occurring in the same sentence and investigate the role of commonsense knowledge in modeling the relation between targets.

Acknowledgements

This work was conducted within the Rolls-Royce@NTU Corp Lab with support from the National Research Foundation Singapore under the Corp Lab@University Scheme.

References

Ahn, S.; Choi, H.; Pärnamaa, T.; and Bengio, Y. 2016. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* abs/1409.0473.

Cambria, E.; Fu, J.; Bisio, F.; and Poria, S. 2015. AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, 508–514.

Cambria, E.; Poria, S.; Bajpai, R.; and Schuller, B. 2016. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *COLING*, 2666–2677.

Cambria, E.; Das, D.; Bandyopadhyay, S.; and Feraco, A. 2017a. *A Practical Guide to Sentiment Analysis*. Cham, Switzerland: Springer.

Cambria, E.; Poria, S.; Gelbukh, A.; and Thelwall, M. 2017b. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems* 32(6):74–80.

Chaturvedi, I.; Ragusa, E.; Gastaldo, P.; Zunino, R.; and Cambria, E. 2017. Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*.

Chen, P.; Sun, Z.; Bing, L.; and Yang, W. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP 2017*, 463–472.

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 *the 52nd ACL*, 49–54.

He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *the 25th WWW*, 507–517.

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

Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval 2014*, 437–442.

Lakkaraju, H.; Socher, R.; and Manning, C. 2014. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*.

Lo, S. L.; Cambria, E.; Chiong, R.; and Cornforth, D. 2017. Multilingual sentiment analysis: From formal to informal and scarce resource languages. *Artificial Intelligence Review* 48(4):499–527.

Ma, Y.; Cambria, E.; and Gao, S. 2016. Label embedding for zero-shot fine-grained named entity typing. In *COLING*, 171–180.

Majumder, N.; Poria, S.; Gelbukh, A.; and Cambria, E. 2017. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems* 32(2):74–79.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*, 3111–3119.

Nakashole, N., and Mitchell, T. M. 2015. A knowledge-intensive model for prepositional phrase attachment. In *ACL (1)*, 365–375.

Nguyen, T. H., and Shirai, K. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *EMNLP 2015*, 2509–2514.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *the 8th SemEval 2014*, 27–35.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; and Androutsopoulos, I. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *the 9th SemEval*, 486–495.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; Hoste, V.; Apidianaki, M.; Tannier, X.; Loukachevitch, N.; Kotelnikov, E.; Bel, N.; Jiménez-Zafra, S. M.; and Eryiğit, G. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *the 10th SemEval*, 19–30.

Poria, S.; Cambria, E.; Hazarika, D.; and Vij, P. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. In *COLING*, 1601–1612.

Poria, S.; Cambria, E.; Bajpai, R.; and Hussain, A. 2017. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion* 37:98–125.

Poria, S.; Cambria, E.; and Gelbukh, A. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108:42–49.

Rahman, A., and Ng, V. 2011. Coreference resolution with world knowledge. In *the 49th ACL*, 814–824.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *the 13th CoNLL*, 147–155.

Saeidi, M.; Bouchard, G.; Liakata, M.; and Riedel, S. 2016. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *COLING 2016*, 1546–1556.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Effective lstms for target-dependent sentiment classification. In *COLING 2016*, 3298–3307.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *the EMNLP 2016*, 214–224.

Wagner, J.; Arora, P.; Cortes, S.; Barman, U.; Bogdanova, D.; Foster, J.; and Tounsi, L. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *the 8th SemEval*, 223–229.

Wang, Y.; Huang, M.; zhu, x.; and Zhao, L. 2016. Attention-based lstm for aspect-level sentiment classification. In *the EMNLP 2016*, 606–615.

Wang, B.; Liakata, M.; Zubiaga, A.; and Procter, R. 2017. Tdparse: Multi-target-specific sentiment recognition on twitter. In *the 15th EACL*, 483–493.

Xia, Y.; Cambria, E.; Hussain, A.; and Zhao, H. 2015. Word polarity disambiguation using bayesian model and opinion-level features. *Cognitive Computation* 7(3):369–380.

Xu, Z.; Liu, B.; Wang, B.; Sun, C.; and Wang, X. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *arXiv preprint*.

Yang, B., and Mitchell, T. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *the 55th ACL*, 1436–1446.