Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis

Article *in* Cognitive Computation · August 2018

DOI: 10.1007/s12559-018-9549-x

Yukun Ma
Nanyang Technological University
16 PUBLICATIONS **1,343** CITATIONS

SEE PROFILE

Erik Cambria
Nanyang Technological University
456 PUBLICATIONS **33,158** CITATIONS

SEE PROFILE

Amir Hussain
Edinburgh Napier University
646 PUBLICATIONS **15,514** CITATIONS

SEE PROFILE

Cognitive Computation manuscript No.
(will be inserted by the editor)

Sentic LSTM: A Hybrid Network for Targeted Aspect-Based Sentiment Analysis

Yukun Ma · Haiyun Peng · Erik Cambria · Tahir Khan · Amir Hussain

Received: date / Accepted: date

Abstract
Background Sentiment analysis has emerged as one of the most popular NLP tasks in recent years. A classic setting of the task mainly involves classifying the overall sentiment polarity of the inputs. However, it is based on the assumption that the sentiment expressed in a sentence is unified and consistent, which does not hold in the reality. As a fine-grained alternative of the task, analyzing the sentiment towards a specific target and aspect has drawn much attention from the community for its more practical assumption that sentiment is dependent on a particular set of aspects and entities.

Recently, deep neural models have achieved great successes on sentiment analysis. As a functional simulation of the behavior of human brains and one of the most successful deep neural models for sequential data, long short-term memory (LSTM) networks are excellent in learning implicit knowledge from data. However, it is impossible for LSTM to acquire explicit knowledge such as commonsense facts from the training data for accomplishing their specific tasks. On the other hand, emerging knowledge bases have brought a variety of knowledge resources to our attention, and it has been acknowledged that incorporating the background knowledge is an important add-on for many NLP tasks.
*Methods*In this paper, we propose a knowledge-rich solution to targeted aspect-based sentiment analysis with a specific focus on leveraging common-sense knowledge in the deep neural sequential model. To explicitly model the inference of the sentiment dependent we augment the LSTM with a stacked attention mechanism consisting of attention models for target-level sentence-level, respectively. In order to explicitly integrate the explicit knowledge with implicit knowledge, we propose an extension of LSTM, termed Sentic LSTM. The extended LSTM cell includes a separate output gate that interpolates token-level memory and the concept-level input. In addition, we we propose an extension of the Sentic-LSTM by creating a hybrid of the LSTM and a recurrent additive network that simulates the sentic patterns used in our previous work. *Results* In this paper, we are mainly concerned with a joint task combining the target-dependent aspect detection and targeted aspect-based polarity classification. The performance of proposed methods on this joint task is evaluated on two benchmark data sets. The experiment shows that the combination of proposed attention architecture and knowledge-embedded LSTM could outperform state-of-the-art methods in two targeted aspect sentiment tasks.
Conclusion We present a knowledge-rich solution for the task of targeted aspect-based sentiment analysis. Our model can effectively incorporate the commonsense knowledge into the deep neural network and be trained in an end-to-end manner. We show that the two-step attentive neural architecture as well as the proposed Sentic-LSTM and H-Sentic-LSTM can achieve an improved performance on resolving the aspect categories and sentiment polarity for a targeted entity in its context over state-of-the-art systems.

Y. Ma and H. Peng and E. Cambria
School of Computer Science and Engineering
Nanyang Technological University
50 Nanyang Ave, Singapore 639798
E-mail: mayu0010@e.ntu.edu.sg
E-mail: peng0065@w.ntu.edu.sg
E-mail: cambria@ntu.edu.sg
Tahir Khan
Alef Education Consultancy
Abu Dhabi, UAE
E-mail: tahir@alefeducation.com
Amir Hussain
University of Stirling
Stirling FK9 4LA, Scotland, UK
E-mail: ahu@cs.stir.ac.uk

1 Introduction

Sentiment analysis [1] has turned out to be a critical step for processing natural language processing, especially for social media data on online communities, blogs, Wikis, microblogging platforms, and other online collaborative media. As a branch of affective computing research [2], sentiment analysis aims to categorize text (but sometimes also audio and video [3]) into either positive or negative (or neutral [4] in some cases). Early works on sentiment analysis focus on resolving the overall sentiment of a text unit [5, 6], assuming that a sentence expresses a unified sentiment or polarity. In contrast, aspect-based sentiment analysis (ABSA) [7–11] extends the typical setting of sentiment analysis with a more realistic assumption that polarity is associated with specific aspects (or product features). Taking the sentence "The design of the space is good but the service is horrible" for an example, the sentiment expressed towards the two aspects, "space" and "service", is the completely opposite. On the other hand, targeted (or target-dependent) sentiment classification (TSA) [12–14] resolves the sentiment polarity of a given target expression in its context, assuming that sentiment is dependent on targeted entities. For instance, in the sentence "I just log on my [facebook]. [Transformer] is boring", the sentiment expressed towards [Transformer] is negative, while no clear sentiment for [facebook]. Most recently, targeted aspect-based sentiment analysis (TABSA) [15] combines the challenges of ABSA and TSA. Namely, the task requires detection of the aspect category as well as the resolution of the polarity for specific aspects of a given targeted entity. Figure 1 illustrates another example of TABSA in Sentihood dataset[1] [15].

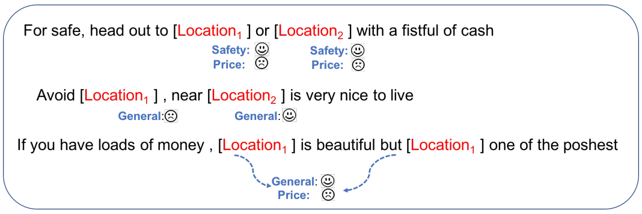
For safe, head out to [Location1] or [Location2] with a fistful of cash
Safety: ☺
Price: ☹
Safety: ☺
Price: ☹
Avoid [Location1] , near [Location2] is very nice to live
General:☹
General:☺
If you have loads of money , [Location1] is beautiful but [Location1] one of the poshest
General: ☺
Price: ☹

Fig. 1: Example of targeted aspect-based sentiment analysis in Sentihood

As a popular and effective solution to many NLP task, deep learning methods [12,14,16–18] have achieved great successes when applied to ABSA or TSA. Especially, neural sequential model, such as LSTM [19], attracts more and more attention for their capacity of representing sequential information. Although it is not scientifically validated that the LSTM is cognitive plausible, the functionality, especially the forget mechanism, of LSTM is considered simulating the functionality of human brains. Due to use of gated functions, LSTM can effectively learn implicit knowledge from sequences by avoiding the troublesome of gradient vanishing. In contrast, explicit knowledge such as commonsense knowledge is hard to be learned from the data where commonsense facts are not explicitly annotated.

We, therefore, target three problems remaining unsolved by current state-of-the-art methods. Firstly, given a target might be composed of multiple instances (mentions of the same target) or multiple words, existing research assumes all instances are of equal importance and simply computes an average vector over instances. This oversimplification conflicts with the fact that part of the target expression is more tightly tied with sentiment than the rest. Secondly, hierarchical attention exploited by existent methods only implicitly model the process of inferring the sentiment evidence as blackbox. Last but not least, existing research falls short in effectively incorporating external sentiment knowledge. The emergence of a variety of knowledge [20, 21] has facilitated many applications of text processing [22–27] and image processing [28, 29] with accessing external information that are not available in the limited training data. Especially, the task of TABSA might benefit from using the affective commonsense knowledge [30] for that the affective properties strongly correlate with aspects and sentiment polarity. To address these problems, we propose a two-step attention architecture with an extended LSTM cell that can better leverage the external knowledge. We identify our contribution as three folds:

1. We propose a two-step attention model which explicitly attends to first the words of target expression, and then to the whole sentence. The two-step attention model simulates the process of first locating and memorizing the targets and then searching for related sentiment cues over the whole sentence.
2. We extend the classic LSTM cell with components accounting for integration with external knowledge. The extended LSTM is capable of utilizing the explicit knowledge to control the information flows from time to time as well as generating the output fed into the classifier.
3. To the best of our knowledge, we are the first to incorporate affective commonsense knowledge into a deep neural network for modeling sequences.

[1] The Sentihood dataset has masked all targeted entities with "Location" + INDEX.

2 Related Work

In this section, we survey several related research lines: ABSA, TSA, TABSA, and finally works on incorporating external knowledge into deep neural models.

2.1 Aspect-based Sentiment Analysis

ABSA is the task to classify 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 the sentences [31,32]. Motivated by the success of deep learning methods in representation learning, many recent works [13,16,17,33] utilize deep neural networks, such as LSTM, 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 [17], 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 attention mechanism for it overcomes the shortcoming of recurrent neural network (RNN) that suffers from information loss when only one single output (e.g., the output at the end of the sequence) is used by the classifier.

2.2 Targeted Sentiment Analysis

In comparison with the ABSA, targeted sentiment analysis aims to analyze the sentiment with regard to targeted entities in the sentence. It is thus critical for targeted sentiment analysis methods, e.g., the target-dependent LSTM model (TD-LSTM) and target connection LSTM model [12], to model the interaction between sentiment targets and the whole sentence. In order to obtain the target-dependent sentence representation, the TD-LSTM directly uses the hidden outputs of a bidirectional-LSTM sentence encoders panning the target mentions, while TC-LSTM extends TD-LSTM 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 network [18] and recurrent attention model [34] have achieved superior performance by learning a deep attention over the single-level attention, for that multiple passes (or hops) over the input sequence could refine the attended words again and again to find the most important words.

These existent approaches have either ignored the problem of multiple target instances (or words) or simply used an averaging vector over target expression [14, 18]. Our method differs with existing methods by weighting each target word with an attention weight so that the given target tends to be represented by its most informative part.

2.3 Targeted Aspect-based Sentiment Analysis

Two baseline systems [15] are proposed together with SentiHood dataset: 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 the TD-LSTM [12] that simply uses the hidden outputs at the position of target instances assuming that all target instances are equally important.

2.4 Incorporating External Knowledge

Existent study on incorporating external knowledge into deep neural networks is also closely related to this work. External knowledge base has been typically used as a source of features [22,35,36]. Most recently, neural sequential models [27,37] 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 the knowledge and the computational structure. Our Sentic LSTM is inspired by [24] which adds a knowledge recall gate to the cell state of LSTM. However, our method differs from [24] in the way of using external knowledge to generate the hidden outputs and controlling the information flow.

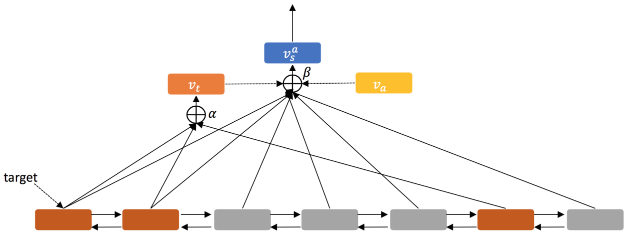

Fig. 2: Overview of the two-step attentive neural architecture

3 Methodology

In this section, we describe the proposed method in detail. We start with briefly introducing the task definition of TABSA followed by an overview of the whole neural architecture. Afterwards, we describe the two-step attention model. Lastly, we describe the proposed knowledge-embedded extension of the LSTM.

Table 1: List of Aspects Defined in SentiHood Dataset

General	Shopping
Price	Multicultural
Transit-location	Green-nature
Safety	Dinning
Live	Quiet
Nightlife	Touristy

3.1 Task Definition

A sentence s is composed of a sequence of words. Similar to [14], we consider all mentions of the same target as a single target. A target t composed of m words (can be either consecutive or not) 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 TABSA can be divided into two subtasks. Firstly, it resolves the aspect categories of t belonging to a predefined set (Table 1). Secondly, it classifies the sentiment polarity with regard 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**].

3.2 Overview

In this section, we provide an overview of our proposed method. Our proposed neural architecture is comprised of two components: the sequence encoder and a hierarchical attention components. Fig. 2 illustrates our neural architecture. 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 $\{v_1, v_2, \cdots, v_L\}$, where L is the length of the sentence. The sequence encoder transforms the word embeddings into a sequence of hidden outputs, on top of which the attention model is built. The target-level attention takes as input the hidden outputs of target expression (highlighted in red) and encodes the salience of each word via a self-attention. The target-level attention model then outputs a weighted sum of these hidden outputs as a vector representation of the given target. At the second step, we feed the target embedding together with aspect embeddings as queries to a sentence-level attention model to transform the whole sentence into a vector. A softmax layer is used for mapping the sentence vector to an output label (e.g., None, Neural, Negative, and Positive for a 4-class setting; or None, Negative, and Positive for a 3-class setting) that jointly represents its sentiment polarity and membership of an aspect.

3.3 Long Short-Term Memory Network

The sentence is encoded using an extension of RNN [38], termed LSTM [19], which was firstly introduced by [19] 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 cell is defined as below:

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

, 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.

3.4 Target-level Attention

Based on the attention mechanism, we calculate an attention vector for a target expression. A target might consist of 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.

3.5 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.,

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

, where the vector $\beta = [\beta_1, \beta_2, \cdots, \beta_L]$ is called sentence-level attention vector. Each element β_i encodes the salience of the word w_i in the sentence s with regard to the aspect a and target T. Existing research on target or aspect-based sentiment analysis 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 $tanh$ activating 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.

Table 2: Example of AffectNet

AffectNet	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	...

3.6 Sentic LSTM

In this paper, we consider using the affective commonsense knowledge as our knowledge source to be embedded into the sequence model. Affective commonsense knowledge such as AffectNet [39] contains concepts associated with a rich set of affective properties (as shown in Table 2). These affective properties provide not only concept-level features but also semantic links to the aspects and their sentiment polarity. For example, the concept 'rotten fish' has property "KindOf-food" which can be directly related to aspects such as 'restaurant' or 'food quality', and properties such as 'Arises-joy' could contribute positively to the classification of sentiment polarity.

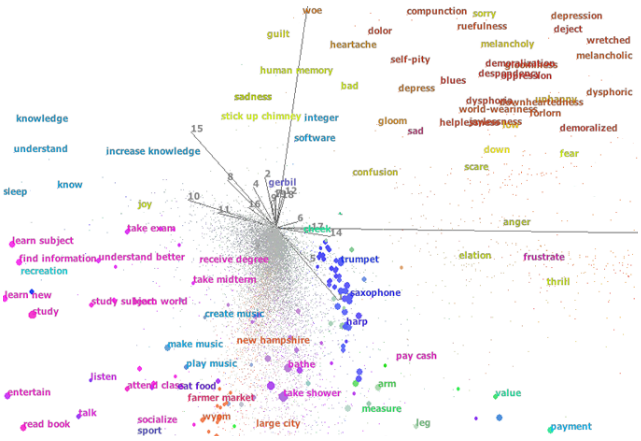

Fig. 3: Visualization of AffectiveSpace

However, AffectNet is of high dimensions which hinders it from being used in deep neural models. AffectiveSpace [30] has been built to map the concepts of AffectNet to continuous low-dimensional embeddings (as shown in Figure 3) without losing the semantic and affective relatedness in the original space. Based on this new space of concepts, we embedded the concept-level information into deep neural sequential models to better classify the aspects and sentiment of sentences.

In order to leverage the commonsense knowledge with efficacy, we propose an extension of LSTM, termed

Sentic LSTM. It is reasonable to assume that the knowledge concepts contains information complementary to the textual word sequence, especially when the knowledge base in use is designed to include abstract concepts. Our 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 in Equation 6.

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

In this paper, as we find that there exists 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 extension of LSTM is illustrated in Equation 7. At first, we assume that the affective concepts are meaningful cues to control the information of token-level information. For example, a multi-word concept 'rotten fish' might indicate the word 'rotten' is a sentiment-related modifier to 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 affected by input tokens conflicted with the pre-existed knowledge. Similarly, the output gate uses the knowledge to filter out irrelevant information stored in the memory. Another important feature of our extension of the 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 AffectiveSpace to the same space as the memory outputs. In other words, o_i^c models the relative contribution 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 [27] that allows the model to choose whether to use the affective knowledge or not.

3.7 Hybrid Sentic-LSTM

Inspired by the work [40], we propose a simplified version of Sentic-LSTM that is a hybrid of LSTM and recurrent additive network. This variant of the Sentic-LSTM could involve the concept-level input to the recurrent connection and maintain a reduced number of parameters as compared with the sentic LSTM. Another intuition is that the additive operation on each concept embedding over time steps simulates inference process done by the rule-based sentic patterns [41] that were used in the previous system. A mathematical description of the hybrid Sentic-LSTM is give in Equation 8.

$$\begin{aligned}
M_i &= W_c \mu_i \\
f_i &= \sigma(W_f[x_i, h_{i-1}, \mu_i] + b_f) \\
I_i &= \sigma(W_I[x_i, h_{i-1}, \mu_i] + b_I) \\
\widetilde{C}_i &= tanh(W_C[x_i, h_{i-1}] + b_C) \\
o_i^c &= \sigma(W_{co}[x_i, h_{i-1}, \mu_i] + b_{co}) \\
C_i &= f_i * C_{i-1} + I_i * \widetilde{C}_i + o_i^c * M_i \\
h_i &= C_i
\end{aligned} \tag{8}$$

Similar to recurrent additive network, we could rewrite the hidden output at time step i as Equation 9.

$$\begin{aligned}
C_i^\mu &= f_i * (C_{i-1} - C_i^\mu) + I_i * \widetilde{C}_i + f_i * C_i^\mu + o_i^c * M_i \\
& I_i * \widetilde{C}_i + f_i * C_i^\mu + o_i^c * M = \sum_{j=0}^{i} w_j^i * M_j
\end{aligned} \tag{9}$$

w_j^i is a product of the input and forget gates. We could see that the hidden output at time step i is a hybrid of a simplified LSTM, whose gates are coupled with both word-level and concept-level input, and a recurrent additive component accumulating the information from the concept-level input over previous time steps.

3.8 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.,

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

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

$$p^a_{c,t} = softmax(W^p v^a_{s,t} + b^a_s)$$

, where W^p and b^a_s 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.

4 Experiment

4.1 Dataset and Resources

We evaluate our method on two datasets: SentiHood dataset [15] and a subset of SemEval 2015 [42]. The Sentihood dataset was built by querying Yahoo! Answers with location names of London city. Table 3 shows statistics of SentiHood datasets. The whole dataset is split into train, test, and development set by the author. 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 2/3 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.

To inject the commonsense knowledge, we use a syntax-based concept parser[2] to extract a set of concept candidates at each time step, and use pre-trained 100-dimensional AffectiveSpace embedding[3] as the concept embeddings. In the case that no concepts are extracted, a zero vector is used as the concept input.

Table 3: 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

4.2 Experiment Setting

We evaluate our method on two sub-tasks of the target-specific aspect-based sentiment analysis: 1) aspect categorization and 2) aspect-based sentiment classification. Following Saeidi et al. [15], we treat the outputs of aspect-based classification as hierarchical classes. For aspect categorization, we output the label with the highest probability for each aspect. The labels are, for example in the 3-class setting, ‘Positive’, ‘Negative’, and ‘None’, where ‘None’ means the aspect should not be bonded to the given target. For aspect-based sentiment classification, we only look at the probability of ‘Positive’ and ‘Negative’, while ignoring the scores of ‘None’[4]. For evaluating the aspect-based sentiment classification, we calculate the accuracy averaged over aspects. We evaluate the aspect categorization as a multi-label classification problem, and the results, therefore, 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 (Strict Acc.).

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 $= \frac{1}{|D|}\sum_{t\in D}\frac{|Y_t\cap\widehat{Y}_t|}{Y_t}$, 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$= \frac{\sum_{t\in D}|Y_t\cap\widehat{Y}_t|}{\sum_{t\in D}Y_t}$.

[2] http://github.com/senticnet/concept-parser

[3] http://sentic.net/downloads

[4] On SemEval 2015, we use ‘Negative’,‘Positive’, ‘Neutral’ and ‘None’.

Table 4: Performance of systems 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**
H-Sentic-LSTM + TA + SA	69.20	**67.52**	78.66	78.10	78.77	**77.87**	87.00	87.78

4.3 Performance Comparison

We compare our proposed method with the methods that have been proposed for TABSA as well as those proposed for ABSA or TSA but applicable to TABSA. 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 the best in development set. For SemEval-2015 dataset, we report the results of the final epoch.

- **TD-LSTM**: This method is the adaptation of TD-LSTM [12, 15]. It 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.
- **Bi-LSTM + TA**: Our method learns an instance attention on top of the outputs of LSTM to model the contribution of each instance.
- **Bi-LSTM + TA + SA**: In addition to target instance attention, we add to the model a sentence-level attention.
- **Bi-LSTM + TA + DMN SA** : The sentence-level attention is replaced by a dynamic memory network with multiple hops [18]. We run the memory network with different numbers of hops and report the result with 4 hops which produces the best performance on development set of Sentihood. We exclude the case of zero hops which correspond 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 [27], which learns an attention over the concept embeddings. The concept embedding is 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 [24].
- **Sentic-LSTM + TA + SA**: The encoder is replaced with the proposed knowledge-embedded LSTM.
- **H-Sentic-LSTM + TA + SA**: The hybrid Sentic-LSTM that has a simplified recurrent additive component accounting for the concept-level input.

The word embedding of input layer is initialized by pre-trained skip-gram model [43] with 150 hidden units on a combination of Yelp[5] and Amazon review dataset [44], and we use 50 hidden units for the bi-directional LSTM.

4.4 Results of Attention Model

Table 4 and Table 5 show the performance on the Sentihood dataset and SemEval-15 dataset respectively. In comparison with the non-attention baseline (TD-LSTM), we can find that our best attention-based model significantly improves the aspect detection by more than 35% and sentiment classification by approximately 10% on the Sentihood dataset. It also achieves similar improvement of sentiment classification on Semeval Dataset. However, it is notable that, on the SemEval-2015 dataset, the improvement of aspect detection is relatively smaller. We conjecture the reason is that Sentihood dataset has masked the target as a special word "LOCATION", which proved 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 dataset. Although not significant, the target-level attention performs better than target averaging model (i.e., TD-LSTM) for that 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 achieve significant improvements on both aspect categorization

[5] `http://yelp.com.sg/dataset/challenge`

Table 5: Performance of systems 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**
H-Sentic-LSTM + TA + SA	**69.19**	**77.55**	**75.00**	74.11

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 produce any improvement. The performance even falls down significantly on SemEval-2015 dataset 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, making it less applicable to small and sparse dataset such as SemEval 2015.

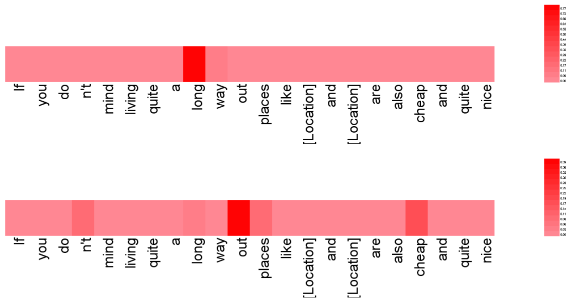

Fig. 4: Example of sentence-level attention

4.5 Visualization of Attention

We visualize the attention vectors of sentence-level attention in Figure 4 with regard to “Transition-location” and “Price” aspects. The two attention vectors have encoded quite different concerns in the word sequence. In the first example, the ‘Transition-location’ attention attends to the words “long” which is expressing a negative sentiment towards the target. In comparison, the ‘Price’ attention attends the more to the word ‘cheap’ that 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 **??**, when there are multiple target instances (or words), the target-level attention is capable of selecting the first target instance which is tied with the clearer sentiment.

4.6 The Result of Knowledge-embedded LSTM

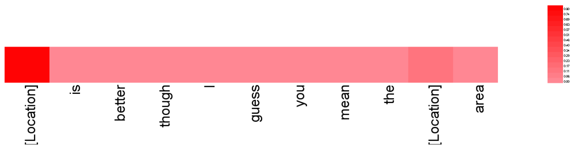

Fig. 5: Example of target-level attention

It can be seen from Table 4 and 5 that injecting the knowledge into the model improves the performance in general. Since the affective space used in our experiment contains the information of affective properties that are semantically related to the aspects and sentiment polarity, it is reasonable to find that it improves performance on both tasks. The results also show that our proposed Sentic LSTM outperforms baseline knowledge-rich methods, even though not very significantly. Comparing Sentic LSTM with KB feats that uses extracted concepts as features, we can find that Sentic LSTM improves more on the aspect categorization part, indicating that advantage of using a knowledge output gate to choose between commonsense knowledge and inner memory. The superior performance of Sentic LSTM over Recall-LSTM and KBA indicates that the triggered knowledge concepts can also help to filter the information that conflicts with the commonsense.

As compared to the standard Sentic-LSTM, the hybrid of Sentic-LSTM and recurrent additive neural network (i.e., H-Sentic-LSTM) achieves comparable performance with smaller number of parameters. We notice

Table 6: Comparison of systems using AffectiveSpace and SentiWordNet (SH stands for Sentihood, and SE stands for SemEval-15)

	Aspect Categorization						Sentiment	
	Strict Acc. (%)		Macro F1 (%)		Micro F1 (%)		Sentiment Acc. (%)	
	SH	SE	SH	SE	SH	SE	SH	SE
AffectiveSpace	**67.71**	**70.47**	**78.05**	**77.90**	**78.19**	**75.12**	**89.63**	**78.31**
SentiWordNet	60.23	66.42	70.03	73.60	69.48	71.18	85.01	75.79

that using the recurrent additive operation on concept-level inputs improves the performance of aspect categorization on SemEval data set, indicating that the additive operation help accumulating the evidence from a broader context, and the generality of the model could possibly benefit more from the reduced number of parameters given that size of training data is much smaller than Sentihood data.

On the other hand, the performance of H-Sentic-LSTM on resolving the polarity of aspect-target pairs are slightly under-performing Sentic-LSTM. Since the additive operation is only able to introduce non-negative contribution from previous step, we conjecture the decrease of performance might be a result of the additive operation does not suffice to simulate complicated sentiment patterns such as negation.

4.7 AffectiveSpace versus SentiWordNet

At last, we compare systems using different sentiment knowledge bases. SentiWordNet is a lexicon-based sentiment knowledge base consisting of word sense synsets annotated with sentiment polarity. However, it is notable that SentiWordNet contains neither commonsense concepts nor affective properties, which are the key features of AffectiveSpace. Consequently, we have to use randomly initialized embeddings for representing the SentiWordNet synsets. Word synsets are mapped to the same 100 dimension embeddings as AffectiveSpace does. Each word in the sentence is mapped to its word sense with the help of a Word Sense Disambiguation tool. We deliberately remove the neutral synsets (i.e., those having zero values for both Positive and Negative) to emphasize on the sentiment-bearing words. Table 6 shows the comparison of the two knowledge base. We report the results using our overall best-performed system (Sentic LSTM + TA + SA). It shows that using the AffectiveSpace achieves superior performance than using SentiWordNet. We conjecture the reason is that the word sense synsets is not as informative as affective properties. Moreover, probably because the link between word senses and aspects are not straightforward, we find the gap in sentiment classification is smaller than aspect categorization.

5 Conclusion

In this paper, we propose a neural architecture together with two extension of the standard LSTM for the task of targeted aspect-based sentiment analysis. We explicitly model the process of inferring the sentiment aspects and polarity as a two-step attention model that encodes the target and full sentence in sequence. The target-level attention attends to the sentiment-salient part of a target expression and generates a more accurate representation of the target, while the aspect-target-dependent sentence-level attention searches for the target and aspect dependent evidence over the full sentence. Moreover, we validate the efficacy of the proposed extensions of the LSTM cell as well as the benefit of using affective commonsense knowledge for the task of targeted aspect-based sentiment analysis. In the future, we would like to take into account the relations between concepts when performing the task.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Informed Consent Informed consent was not required as no human or animals were involved.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. *A Practical Guide to Sentiment Analysis*. Springer, Cham, Switzerland, 2017.
2. Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.
3. Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Mazumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *ACL*, pages 873–883, 2017.

4. Iti Chaturvedi, Edoardo Ragusa, Paolo Gastaldo, Rodolfo Zunino, and Erik Cambria. Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*, 2017.
5. Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388, 2007.
6. Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 341–349, New York, NY, USA, 2002. ACM.
7. Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.
8. Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.
9. Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, 2016.
10. Yunqing Xia, Erik Cambria, and Amir Hussain. Aspnet: Aspect extraction by bootstrapping generalization and propagation using an aspect network. *Cognitive Computation*, 7(2):241–253, 2015.
11. Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Federica Bisio. Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *IJCNN*, pages 4465–4473, 2016.
12. Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
13. Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
14. Bo Wang, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. Tdparse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 483–493, Valencia, Spain, April 2017. Association for Computational Linguistics.
15. Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
16. Thien Hai Nguyen and Kiyoaki Shirai. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
17. Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas, November 2016. Association for Computational Linguistics.
18. Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224, Austin, Texas, November 2016. Association for Computational Linguistics.
19. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
20. Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, Valletta, Malta, 2010. European Language Resources Association (ELRA).
21. Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Bjoern Schuller. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2666–2677, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
22. Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.
23. Yukun Ma, Jung-jae Kim, Benjamin Bigot, and Tahir Muhammad Khan. Feature-enriched word embeddings for named entity recognition in open-domain conversations. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6055–6059. IEEE, 2016.
24. Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *arXiv preprint arXiv:1605.05110*, 2016.
25. Yang Li, Quan Pan, Tao Yang, Suhang Wang, Jiliang Tang, and Erik Cambria. Learning word representations for sentiment analysis. *Cognitive Computation*, pages 1–9, 2017.
26. Nir Ofek, Soujanya Poria, Lior Rokach, Erik Cambria, Amir Hussain, and Asaf Shabtai. Unsupervised commonsense knowledge enrichment for domain-specific sentiment analysis. *Cognitive Computation*, 8(3):467–477, 2016.
27. Bishan Yang and Tom Mitchell. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada, July 2017. Association for Computational Linguistics.
28. Erik Cambria and Amir Hussain. Sentic album: content-, concept-, and context-based online personal photo man-

agement system. *Cognitive Computation*, 4(4):477–496, 2012.

29. Qiu-Feng Wang, Erik Cambria, Cheng-Lin Liu, and Amir Hussain. Common sense knowledge for handwritten chinese text recognition. *Cognitive Computation*, 5(2):234–242, 2013.
30. Erik Cambria, Jie Fu, Federica Bisio, and Soujanya Poria. Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, pages 508–514, 2015.
31. Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.
32. Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.
33. Himabindu Lakkaraju, Richard Socher, and Chris Manning. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*. Curran Associates, Inc., 2014.
34. Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 463–472, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
35. Altaf Rahman and Vincent Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 814–824. Association for Computational Linguistics, 2011.
36. Ndapandula Nakashole and Tom M Mitchell. A knowledge-intensive model for prepositional phrase attachment. In *ACL (1)*, pages 365–375, 2015.
37. Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*, 2016.
38. Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
39. Luca Oneto, Federica Bisio, Erik Cambria, and Davide Anguita. Semi-supervised learning for affective commonsense reasoning. *Cognitive Computation*, 9(1):18–42, 2017.
40. Kenton Lee, Omer Levy, and Luke Zettlemoyer. Recurrent additive networks. *arXiv preprint arXiv:1705.07393*, 2017.
41. Soujanya Poria, Erik Cambria, Gregoire Winterstein, and Guang-Bin Huang. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69:45–63, 2014.
42. Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado, June 2015. Association for Computational Linguistics.
43. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
44. Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.