Target-Sensitive Memory Networks for Aspect Sentiment Classification

Shuai Wang[†], **Sahisnu Mazumder**[†], **Bing Liu**[†], **Mianwei Zhou**[‡], **Yi Chang**[§]
[†]Department of Computer Science, University of Illinois at Chicago, USA
[‡]Plus.AI, USA
[§]Artificial Intelligence School, Jilin University, China
shuaiwanghk@gmail.com, sahisnumazumder@gmail.com
liub@uic.edu, mianwei.zhou@gmail.com, yichang@acm.org

Abstract

Aspect sentiment classification (ASC) is a fundamental task in sentiment analysis. Given an aspect/target and a sentence, the task classifies the sentiment polarity expressed on the target in the sentence. Memory networks (MNs) have been used for this task recently and have achieved state-of-the-art results. In MNs, attention mechanism plays a crucial role in detecting the sentiment context for the given target. However, we found an important problem with the current MNs in performing the ASC task. Simply improving the attention mechanism will not solve it. The problem is referred to as *target-sensitive sentiment*, which means that the sentiment polarity of the (detected) context is dependent on the given target and it cannot be inferred from the context alone. To tackle this problem, we propose the *target-sensitive memory networks* (TMNs). Several alternative techniques are designed for the implementation of TMNs and their effectiveness is experimentally evaluated.

1 Introduction

Aspect sentiment classification (ASC) is a core problem of sentiment analysis (Liu, 2012). Given an aspect and a sentence containing the aspect, ASC classifies the sentiment polarity expressed in the sentence about the aspect, namely, positive, neutral, or negative. Aspects are also called *opinion targets* (or simply *targets*), which are usually product/service features in customer reviews. In this paper, we use aspect and target interchangeably. In practice, aspects can be specified by the user or extracted automatically using an aspect extraction technique (Liu, 2012). In this work, we assume the aspect terms are given and only focus on the classification task.

Due to their impressive results in many NLP tasks (Deng et al., 2014), neural networks have been applied to ASC (see the survey (Zhang et al., 2018)). Memory networks (MNs), a type of neural networks which were first proposed for question answering (Weston et al., 2015; Sukhbaatar et al., 2015), have achieved the state-of-the-art results in ASC (Tang et al., 2016). A key factor for their success is the attention mechanism. However, we found that using existing MNs to deal with ASC has an important problem and simply relying on attention modeling cannot solve it. That is, their performance degrades when the sentiment of a context word is sensitive to the given target.

Let us consider the following sentences:

(1) *The screen resolution is* ***excellent*** *but the price is* ***ridiculous***.
(2) *The screen resolution is* ***excellent*** *but the price is* ***high***.
(3) *The price is* ***high***.
(4) *The screen resolution is* ***high***.

In sentence (1), the sentiment expressed on aspect *screen resolution* (or *resolution* for short) is positive, whereas the sentiment on aspect *price* is negative. For the sake of predicting correct sentiment, a crucial step is to first detect the sentiment context about the given aspect/target. We call this step *targeted-context detection*. Memory networks (MNs) can deal with this step quite well because the sentiment context of a given aspect can be captured by the internal attention mechanism in MNs. Concretely, in sentence (1) the word "excellent" can be identified as the sentiment context when *resolution* is specified. Likewise, the context word "ridiculous" will be placed with a high attention when *price* is the target. With the correct targeted-context detected, a trained MN, which recognizes "excellent" as positive sentiment and "ridiculous" as negative sentiment, will infer correct sentiment polarity for the given target. This

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 957–967
Melbourne, Australia, July 15 - 20, 2018.

is relatively easy as "excellent" and "ridiculous" are both target-independent sentiment words, i.e., the words themselves already indicate clear sentiments.

As illustrated above, the attention mechanism addressing the targeted-context detection problem is very useful for ASC, and it helps classify many sentences like sentence (1) accurately. This also led to existing and potential research in improving attention modeling (discussed in Section 5). However, we observed that simply focusing on tackling the target-context detection problem and learning better attention are not sufficient to solve the problem found in sentences (2), (3) and (4).

Sentence (2) is similar to sentence (1) except that the (sentiment) context modifying aspect/target *price* is "high". In this case, when "high" is assigned the correct attention for the aspect *price*, the model also needs to capture the sentiment interaction between "high" and *price* in order to identify the correct sentiment polarity. This is not as easy as sentence (1) because "high" itself indicates no clear sentiment. Instead, its sentiment polarity is dependent on the given target.

Looking at sentences (3) and (4), we further see the importance of this problem and also why relying on attention mechanism alone is insufficient. In these two sentences, sentiment contexts are both "high" (i.e., same attention), but sentence (3) is negative and sentence (4) is positive simply because their target aspects are different. Therefore, focusing on improving attention will not help in these cases. We will give a theoretical insight about this problem with MNs in Section 3.

In this work, we aim to solve this problem. To distinguish it from the aforementioned targeted-context detection problem as shown by sentence (1), we refer to the problem in (2), (3) and (4) as the *target-sensitive sentiment* (or target-dependent sentiment) problem, which means that the sentiment polarity of a detected/attended context word is conditioned on the target and cannot be directly inferred from the context word alone, unlike "excellent" and "ridiculous". To address this problem, we propose *target-sensitive memory networks* (TMNs), which can capture the sentiment interaction between targets and contexts. We present several approaches to implementing TMNs and experimentally evaluate their effectiveness.

2 Memory Network for ASC

This section describes our basic memory network for ASC, also as a background knowledge. It does not include the proposed target-sensitive sentiment solutions, which are introduced in Section 4. The model design follows previous studies (Sukhbaatar et al., 2015; Tang et al., 2016) except that a different attention alignment function is used (shown in Eq. 1). Their original models will be compared in our experiments as well. The definitions of related notations are given in Table 1.

t	a target word, $t \in \mathbb{R}^{V\times 1}$
v_t	target embedding of t, $v_t \in \mathbb{R}^{d\times 1}$
x_i	a context word in a sentence, $x_i \in \mathbb{R}^{V\times 1}$
m_i, c_i	input, output context embedding of word x_i, and $m_i, c_i \in \mathbb{R}^{d\times 1}$
V	number of words in vocabulary
d	vector/embedding dimension
A	input embedding matrix $A \in \mathbb{R}^{d\times V}$
C	output embedding matrix $C \in \mathbb{R}^{d\times V}$
α	attention distribution in a sentence
α_i	attention of context word i, $\alpha_i \in (0, 1)$
o	output representation, $o \in \mathbb{R}^{d\times 1}$
K	number of sentiment classes
s	sentiment score, $s \in \mathbb{R}^{K\times 1}$
y	sentiment probability

Table 1: Definition of Notations

Input Representation: Given a target aspect t, an embedding matrix A is used to convert t into a vector representation, v_t ($v_t = At$). Similarly, each context word (non-aspect word in a sentence) $x_i \in \{x_1, x_2, ...x_n\}$ is also projected to the continuous space stored in memory, denoted by m_i ($m_i = Ax_i$) $\in \{m_1, m_2, ...m_n\}$. Here n is the number of words in a sentence and i is the word position/index. Both t and x_i are one-hot vectors. For an aspect expression with multiple words, its aspect representation v_t is the averaged vector of those words (Tang et al., 2016).

Attention: Attention can be obtained based on the above input representation. Specifically, an attention weight α_i for the context word x_i is computed based on the alignment function:

$$\alpha_i = softmax(v_t^T M m_i) \quad (1)$$

where $M \in \mathbb{R}^{d\times d}$ is the general learning matrix suggested by Luong et al. (2015). In this manner, attention $\alpha = \{\alpha_1, \alpha_2, ..\alpha_n\}$ is represented as a vector of probabilities, indicating the weight/importance of context words towards a given target. Note that $\alpha_i \in (0, 1)$ and $\sum_i \alpha_i = 1$.

Output Representation: Another embedding matrix C is used for generating the individual (output) continuous vector c_i ($c_i = Cx_i$) for each context word x_i. A final response/output vector o is produced by summing over these vectors weighted with the attention α, i.e., $o = \sum_i \alpha_i c_i$.

Sentiment Score (or Logit): The aspect sentiment scores (also called logits) for positive, neutral, and negative classes are then calculated, where a sentiment-specific weight matrix $W \in \mathbb{R}^{K \times d}$ is used. The sentiment scores are represented in a vector $s \in \mathbb{R}^{K \times 1}$, where K is the number of (sentiment) classes, which is 3 in ASC.

$$s = W(o + v_t) \tag{2}$$

The final sentiment probability y is produced with a $softmax$ operation, i.e., $y = softmax(s)$.

3 Problem of the above Model for Target-Sensitive Sentiment

This section analyzes the problem of target-sensitive sentiment in the above model. The analysis can be generalized to many existing MNs as long as their improvements are on attention α only. We first expand the sentiment score calculation from Eq. 2 to its individual terms:

$$\begin{aligned} s = W(o + v_t) &= W(\sum_i \alpha_i c_i + v_t) \\ &= \alpha_1 W c_1 + \alpha_2 W c_2 + ...\alpha_n W c_n + W v_t \end{aligned} \tag{3}$$

where "+" denotes element-wise summation. In Eq. 3, $\alpha_i W c_i$ can be viewed as the individual sentiment logit for a context word and $W v_t$ is the sentiment logit of an aspect. They are linearly combined to determine the final sentiment score s. This can be problematic in ASC. First, an aspect word often expresses no sentiment, for example, "screen". However, if the aspect term v_t is simply removed from Eq. 3, it also causes the problem that the model cannot handle target-dependent sentiment. For instance, the sentences (3) and (4) in Section 1 will then be treated as identical if their aspect words are not considered. Second, if an aspect word is considered and it directly bears some positive or negative sentiment, then when an aspect word occurs with different context words for expressing opposite sentiments, a contradiction can be resulted from them, especially in the case that the context word is a target-sensitive sentiment word. We explain it as follows.

Let us say we have two target words *price* and *resolution* (denoted as p and r). We also have two possible context words "high" and "low" (denoted as h and l). As these two sentiment words can modify both aspects, we can construct four snippets "high price", "low price", "high resolution" and "low resolution". Their sentiments are negative, positive, positive, and negative respectively. Let us set W to $\mathbb{R}^{1 \times d}$ so that s becomes a 1-dimensional sentiment score indicator. $s > 0$ indicates a positive sentiment and $s < 0$ indicates a negative sentiment. Based on the above example snippets or phrases we have four corresponding inequalities: (a) $W(\alpha_h c_h + v_p) < 0$, (b) $W(\alpha_l c_l + v_p) > 0$, (c) $W(\alpha_h c_h + v_r) > 0$ and (d) $W(\alpha_l c_l + v_r) < 0$. We can drop all α terms here as they all equal to 1, i.e., they are the only context word in the snippets to attend to (the target words are not contexts). From (a) and (b) we can infer (e) $Wc_h < -Wv_p < Wc_l$. From (c) and (d) we can infer (f) $Wc_l < -Wv_r < Wc_h$. From (e) and (f) we have (g) $Wc_h < Wc_l < Wc_h$, which is a contradiction.

This contradiction means that MNs cannot learn a set of parameters W and C to correctly classify the above four snippets/sentences at the same time. This contradiction also generalizes to real-world sentences. That is, although real-world review sentences are usually longer and contain more words, since the attention mechanism makes MNs focus on the most important sentiment context (the context with high α_i scores), the problem is essentially the same. For example, in sentences (2) and (3) in Section 1, when *price* is targeted, the main attention will be placed on "high". For MNs, these situations are nearly the same as that for classifying the snippet "high price". We will also show real examples in the experiment section.

One may then ask whether improving attention can help address the problem, as α_i can affect the final results by adjusting the sentiment effect of the context word via $\alpha_i W c_i$. This is unlikely, if not impossible. First, notice that α_i is a scalar ranging in (0,1), which means it essentially assigns higher or lower weight to increase or decrease the sentiment effect of a context word. It cannot change the intrinsic sentiment orientation/polarity of the context, which is determined by Wc_i. For example, if Wc_i assigns the context word "high" a positive sentiment ($Wc_i > 0$), α_i will not make it negative (i.e., $\alpha_i W c_i < 0$ cannot be achieved by chang-

ing α_i). Second, other irrelevant/unimportant context words often carry no or little sentiment information, so increasing or decreasing their weights does not help. For example, in the sentence "the price is high", adjusting the weights of context words "the" and "is" will neither help solve the problem nor be intuitive to do so.

4 The Proposed Approaches

This section introduces six (6) alternative target-sensitive memory networks (TMNs), which all can deal with the target-sensitive sentiment problem. Each of them has its characteristics.

Non-linear Projection (NP): This is the first approach that utilizes a non-linear projection to capture the interplay between an aspect and its context. Instead of directly following the common linear combination as shown in Eq. 3, we use a non-linear projection (*tanh*) as the replacement to calculate the aspect-specific sentiment score.

$$s = W \cdot tanh(\sum_i \alpha_i c_i + v_t) \quad (4)$$

As shown in Eq. 4, by applying a non-linear projection over attention-weighted c_i and v_t, the context and aspect information are coupled in a way that the final sentiment score cannot be obtained by simply summing their individual contributions (compared with Eq. 3). This technique is also intuitive in neural networks. However, notice that by using the non-linear projection (or adding more sophisticated hidden layers) over them in this way, we sacrifice some interpretability. For example, we may have difficulty in tracking how each individual context word (c_i) affects the final sentiment score s, as all context and target representations are coupled. To avoid this, we can use the following five alternative techniques.

Contextual Non-linear Projection (CNP): Despite the fact that it also uses the non-linear projection, this approach incorporates the interplay between a context word and the given target into its (output) context representation. We thus name it Contextual Non-linear Projection (CNP).

$$s = W \sum_i \alpha_i \cdot tanh(c_i + v_t) \quad (5)$$

From Eq. 5, we can see that this approach can keep the linearity of attention-weighted context aggregation while taking into account the aspect information with non-linear projection, which works in a different way compared to NP. If we define $\tilde{c}_i = tanh(c_i + v_t)$, $\tilde{c}_i$ can be viewed as the target-aware context representation of context x_i and the final sentiment score is calculated based on the aggregation of such $\tilde{c}_i$. This could be a more reasonable way to carry the aspect information rather than simply summing the aspect representation (Eq. 3).

However, one potential disadvantage is that this setting uses the same set of vector representations (learned by embeddings C) for multiple purposes, i.e., to learn output (context) representations and to capture the interplay between contexts and aspects. This may degenerate its model performance when the computational layers in memory networks (called "hops") are deep, because too much information is required to be encoded in such cases and a sole set of vectors may fail to capture all of it.

To overcome this, we suggest the involvement of an additional new set of embeddings/vectors, which is exclusively designed for modeling the sentiment interaction between an aspect and its context. The key idea is to decouple different functioning components with different representations, but still make them work jointly. The following four techniques are based on this idea.

Interaction Term (IT): The third approach is to formulate explicit target-context sentiment interaction terms. Different from the targeted-context detection problem which is captured by attention (discussed in Section 1), here the ***target-context sentiment (TCS) interaction*** measures the sentiment-oriented interaction effect between targets and contexts, which we refer to as *TCS interaction* (or *sentiment interaction*) for short in the rest of this paper. Such sentiment interaction is captured by a new set of vectors, and we thus also call such vectors *TCS vectors*.

$$s = \sum_i \alpha_i (W_s c_i + w_I \langle d_i, d_t \rangle) \quad (6)$$

In Eq. 6, $W_s \in \mathbb{R}^{K \times d}$ and $w_I \in \mathbb{R}^{K \times 1}$ are used instead of W in Eq. 3. W_s models the direct sentiment effect from c_i while w_I works with d_i and d_t together for learning the TCS interaction. d_i and d_t are TCS vector representations of context x_i and aspect t, produced from a new embedding matrix D, i.e., $d_i = Dx_i, d_t = Dt$ ($D \in \mathbb{R}^{d \times V}$ and $d_i, d_t \in \mathbb{R}^{d \times 1}$).

Unlike input and output embeddings A and C, D is designed to capture the sentiment interac-

tion. The vectors from D affect the final sentiment score through $w_I\langle d_i, d_t\rangle$, where w_I is a sentiment-specific vector and $\langle d_i, d_t\rangle \in \mathbb{R}$ denotes the dot product of the two TCS vectors d_i and d_t. Compared to the basic MNs, this model can better capture target-sensitive sentiment because the interactions between a context word h and different aspect words (say, p and r) can be different, i.e., $\langle d_h, d_p\rangle \neq \langle d_h, d_r\rangle$.

The key advantage is that now the sentiment effect is explicitly dependent on its target and context. For example, $\langle d_h, d_p\rangle$ can help shift the final sentiment to negative and $\langle d_h, d_r\rangle$ can help shift it to positive. Note that α is still needed to control the importance of different contexts. In this manner, targeted-context detection (attention) and TCS interaction are jointly modeled and work together for sentiment inference. The proposed techniques introduced below also follow this core idea but with different implementations or properties. We thus will not repeat similar discussions.

Coupled Interaction (CI): This proposed technique associates the TCS interaction with an additional set of context representation. This representation is for capturing the global correlation between context and different sentiment classes.

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t\rangle e_i) \tag{7}$$

Specifically, e_i is another output representation for x_i, which is coupled with the sentiment interaction factor $\langle d_i, d_t\rangle$. For each context word x_i, e_i is generated as $e_i = Ex_i$ where $E \in \mathbb{R}^{d\times V}$ is an embedding matrix. $\langle d_i, d_t\rangle$ and e_i function together as a target-sensitive context vector and are used to produce sentiment scores with W_I ($W_I \in \mathbb{R}^{K\times d}$).

Joint Coupled Interaction (JCI): A natural variant of the above model is to replace e_i with c_i, which means to learn a joint output representation. This can also reduce the number of learning parameters and simplify the CI model.

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t\rangle c_i) \tag{8}$$

Joint Projected Interaction (JPI): This model also employs a unified output representation like JCI, but a context output vector c_i will be projected to two different continuous spaces before sentiment score calculation. To achieve the goal, two projection matrices W_1, W_2 and the non-linear projection function $tanh$ are used. The intuition is that, when we want to reduce the (embedding) parameters and still learn a joint representation, two different sentiment effects need to be separated in different vector spaces. The two sentiment effects are modeled as two terms:

$$\begin{aligned} s = &\sum_i \alpha_i W_J \tanh(W_1 c_i) \\ &+ \sum_i \alpha_i W_J \langle d_i, d_t\rangle \tanh(W_2 c_i) \end{aligned} \tag{9}$$

where the first term can be viewed as learning target-independent sentiment effect while the second term captures the TCS interaction. A joint sentiment-specific weight matrix W_J($W_J \in \mathbb{R}^{K\times d}$) is used to control/balance the interplay between these two effects.

Discussions: (a) In IT, CI, JCI, and JPI, their first-order terms are still needed, because not in all cases sentiment inference needs TCS interaction. For some simple examples like "the battery is good", the context word "good" simply indicates clear sentiment, which can be captured by their first-order term. However, notice that the modeling of second-order terms offers additional help in both general and target-sensitive scenarios. (b) TCS interaction can be calculated by other modeling functions. We have tried several methods and found that using the dot product $\langle d_i, d_t\rangle$ or $d_i^T W d_t$ (with a projection matrix W) generally produces good results. (c) One may ask whether we can use fewer embeddings or just use one universal embedding to replace A, C and D (the definition of D can be found in the introduction of IT). We have investigated them as well. We found that merging A and C is basically workable. But merging D and A/C produces poor results because they essentially function with different purposes. While A and C handle targeted-context detection (attention), D captures the TCS interaction. (d) Except NP, we do not apply non-linear projection to the sentiment score layer. Although adding non-linear transformation to it may further improve model performance, the individual sentiment effect from each context will become untraceable, i.e., losing some interpretability. In order to show the effectiveness of learning TCS interaction and for analysis purpose, we do not use it in this work. But it can be flexibly added for specific tasks/analyses that do not require strong interpretability.

Loss function: The proposed models are all trained in an end-to-end manner by minimizing the cross entropy loss. Let us denote a sentence and a

target aspect as x and t respectively. They appear together in a pair format (x, t) as input and all such pairs construct the dataset H. $g_{(x,t)}$ is a one-hot vector and $g^k_{(x,t)} \in \{0, 1\}$ denotes a gold sentiment label, i.e., whether (x, t) shows sentiment k. $y_{x,t}$ is the model-predicted sentiment distribution for (x, t). $y^k_{x,t}$ denotes its probability in class k. Based on them, the training loss is constructed as:

$$loss = -\sum_{(x,t)\in H}\sum_{k\in K} g^k_{(x,t)} \log y^k_{(x,t)} \qquad (10)$$

5 Related Work

Aspect sentiment classification (ASC) (Hu and Liu, 2004), which is different from document or sentence level sentiment classification (Pang et al., 2002; Kim, 2014; Yang et al., 2016), has recently been tackled by neural networks with promising results (Dong et al., 2014; Nguyen and Shirai, 2015) (also see the survey (Zhang et al., 2018)). Later on, the seminal work of using attention mechanism for neural machine translation (Bahdanau et al., 2015) popularized the application of the attention mechanism in many NLP tasks (Hermann et al., 2015; Cho et al., 2015; Luong et al., 2015), including ASC.

Memory networks (MNs) (Weston et al., 2015; Sukhbaatar et al., 2015) are a type of neural models that involve such attention mechanisms (Bahdanau et al., 2015), and they can be applied to ASC. Tang et al. (2016) proposed an MN variant to ASC and achieved the state-of-the-art performance. Another common neural model using attention mechanism is the RNN/LSTM (Wang et al., 2016).

As discussed in Section 1, the attention mechanism is suitable for ASC because it effectively addresses the targeted-context detection problem. Along this direction, researchers have studied more sophisticated attentions to further help the ASC task (Chen et al., 2017; Ma et al., 2017; Liu and Zhang, 2017). Chen et al. (2017) proposed to use a recurrent attention mechanism. Ma et al. (2017) used multiple sets of attentions, one for modeling the attention of aspect words and one for modeling the attention of context words. Liu and Zhang (2017) also used multiple sets of attentions, one obtained from the left context and one obtained from the right context of a given target. Notice that our work does not lie in this direction. Our goal is to solve the target-sensitive sentiment and to capture the TCS interaction, which is a different problem. This direction is also finer-grained, and none of the above works addresses this problem. Certainly, both directions can improve the ASC task. We will also show in our experiments that our work can be integrated with an improved attention mechanism.

To the best of our knowledge, none of the existing studies addresses the target-sensitive sentiment problem in ASC under the purely data-driven and supervised learning setting. Other concepts like sentiment shifter (Polanyi and Zaenen, 2006) and sentiment composition (Moilanen and Pulman, 2007; Choi and Cardie, 2008; Socher et al., 2013) are also related, but they are not learned automatically and require rule/patterns or external resources (Liu, 2012). Note that our approaches do not rely on handcrafted patterns (Ding et al., 2008; Wu and Wen, 2010), manually compiled sentiment constraints and review ratings (Lu et al., 2011), or parse trees (Socher et al., 2013).

6 Experiments

We perform experiments on the datasets of SemEval Task 2014 (Pontiki et al., 2014), which contain online reviews from domain *Laptop* and *Restaurant*. In these datasets, aspect sentiment polarities are labeled. The training and test sets have also been provided. Full statistics of the datasets are given in Table 2.

Dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128

Table 2: Statistics of Datasets

6.1 Candidate Models for Comparison

MN: The classic end-to-end memory network (Sukhbaatar et al., 2015).
AMN: A state-of-the-art memory network used for ASC (Tang et al., 2016). The main difference from MN is in its attention alignment function, which concatenates the distributed representations of the context and aspect, and uses an additional weight matrix for attention calculation, following the method introduced in (Bahdanau et al., 2015).
BL-MN: Our basic memory network presented in Section 2, which does not use the proposed techniques for capturing target-sensitive sentiments.
AE-LSTM: RNN/LSTM is another popular attention based neural model. Here we compare

with a state-of-the-art attention-based LSTM for ASC, AE-LSTM (Wang et al., 2016).

ATAE-LSTM: Another attention-based LSTM for ASC reported in (Wang et al., 2016).

Target-sensitive Memory Networks (TMNs): The six proposed techniques, NP, CNP, IT, CI, JCI, and JPI give six target-sensitive memory networks.

Note that other non-neural network based models like SVM and neural models without attention mechanism like traditional LSTMs have been compared and reported with inferior performance in the ASC task (Dong et al., 2014; Tang et al., 2016; Wang et al., 2016), so they are excluded from comparisons here. Also, note that non-neural models like SVMs require feature engineering to manually encode aspect information, while this work aims to improve the aspect representation learning based approaches.

6.2 Evaluation Measure

Since we have a three-class classification task (positive, negative and neutral) and the classes are imbalanced as shown in Table 2, we use F1-score as our evaluation measure. We report both F1-Macro over all classes and all individual class-based F1 scores. As our problem requires fine-grained sentiment interaction, the class-based F1 provides more indicative information. In addition, we report the accuracy (same as F1-Micro), as it is used in previous studies. However, we suggest using F1-score because accuracy biases towards the majority class.

6.3 Training Details

We use the open-domain word embeddings[1] for the initialization of word vectors. We initialize other model parameters from a uniform distribution U(-0.05, 0.05). The dimension of the word embedding and the size of the hidden layers are 300. The learning rate is set to 0.01 and the dropout rate is set to 0.1. Stochastic gradient descent is used as our optimizer. The position encoding is also used (Tang et al., 2016). We also compare the memory networks in their multiple computational layers version (i.e., multiple hops) and the number of hops is set to 3 as used in the mentioned previous studies. We implemented all models in the TensorFlow environment using same input, embedding size, dropout rate, optimizer, etc. so as to test our hypotheses, i.e., to make sure the achieved improvements do not come from elsewhere. Meanwhile, we can also report all evaluation measures discussed above[2]. 10% of the training data is used as the development set. We report the best results for all models based on their F-1 Macro scores.

6.3.1 Result Analysis

The classification results are shown in Table 3. Note that the candidate models are all based on classic/standard attention mechanism, i.e., without sophisticated or multiple attentions involved. We compare the 1-hop and 3-hop memory networks as two different settings. The top three F1-Macro scores are marked in bold. Based on them, we have the following observations:

1. Comparing the 1-hop memory networks (first nine rows), we see significant performance gains achieved by CNP, CI, JCI, and JPI on both datasets, where each of them has $p < 0.01$ over the strongest baseline (BL-MN) from paired t-test using F1-Macro. IT also outperforms the other baselines while NP has similar performance to BL-MN. This indicates that TCS interaction is very useful, as BL-MN and NP do not model it.
2. In the 3-hop setting, TMNs achieve much better results on Restaurant. JCI, IT, and CI achieve the best scores, outperforming the strongest baseline AMN by 2.38%, 2.18%, and 2.03%. On Laptop, BL-MN and most TMNs (except CNP and JPI) perform similarly. However, BL-MN performs poorly on Restaurant (only better than two models) while TMNs show more stable performance.
3. Comparing all TMNs, we see that JCI works the best as it always obtains the top-three scores on two datasets and in two settings. CI and JPI also perform well in most cases. IT, NP, and CNP can achieve very good scores in some cases but are less stable. We also analyzed their potential issues in Section 4.
4. It is important to note that these improvements are quite large because in many cases sentiment interactions may not be necessary (like sentence (1) in Section 1). The overall good results obtained by TMNs demonstrate their capability of handling both general and target-sensitive sentiments, i.e., the proposed

[1] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[2] Most related studies report accuracy only.

Restaurant						Laptop					
Model	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**	**Model**	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**
MN	58.91	57.07	36.81	82.86	71.52	MN	56.16	47.06	45.81	75.63	61.91
AMN	63.82	61.76	43.56	86.15	75.68	AMN	60.01	52.67	47.89	79.48	66.14
BL-MN	64.34	61.96	45.86	85.19	75.30	BL-MN	62.89	57.16	49.51	81.99	68.90
NP	64.62	64.89	43.21	85.78	75.93	NP	62.63	56.43	49.62	81.83	68.65
CNP	65.58	62.97	47.65	86.12	75.97	CNP	**64.38**	57.92	53.23	81.98	69.62
IT	65.37	65.22	44.44	86.46	76.98	IT	63.07	57.01	50.62	81.58	68.38
CI	**66.78**	65.49	48.32	86.51	76.96	CI	63.65	57.33	52.60	81.02	68.65
JCI	**66.21**	65.74	46.23	86.65	77.16	JCI	**64.19**	58.49	53.69	80.40	68.42
JPI	**66.58**	65.44	47.60	86.71	76.96	JPI	**64.53**	58.62	51.71	83.25	70.06
AE-LSTM	66.45	64.22	49.40	85.73	76.43	AE-LSTM	62.45	55.26	50.35	81.74	68.50
ATAE-LSTM	65.41	66.19	43.34	86.71	76.61	ATAE-LSTM	59.41	55.27	42.15	80.81	67.40
MN (hops)	62.68	60.35	44.57	83.11	72.86	MN (hops)	60.61	55.59	45.94	80.29	66.61
AMN (hops)	66.46	65.57	46.64	87.16	77.27	AMN (hops)	65.16	60.00	52.56	82.91	70.38
BL-MN (hops)	65.71	63.83	46.91	86.39	76.45	BL-MN (hops)	**67.11**	63.10	54.53	83.69	72.15
NP (hops)	65.98	64.18	47.86	85.90	75.73	NP (hops)	**67.79**	63.17	56.27	83.92	72.43
CNP (hops)	66.87	65.32	49.07	86.22	76.65	CNP (hops)	64.85	58.84	53.29	82.43	70.25
IT (hops)	**68.64**	67.11	51.47	87.33	78.55	IT (hops)	66.23	61.43	53.69	83.57	71.37
CI (hops)	**68.49**	64.83	53.03	87.60	78.69	CI (hops)	66.79	61.80	55.30	83.26	71.67
JCI (hops)	**68.84**	66.28	52.06	88.19	78.79	JCI (hops)	**67.23**	61.08	57.49	83.11	71.79
JPI (hops)	67.86	66.72	49.63	87.24	77.95	JPI (hops)	65.16	59.01	54.25	82.20	70.18

Table 3: Results of all models on two datasets. Top three F1-Macro scores are marked in bold. The first nine models are 1-hop memory networks. The last nine models are 3-hop memory networks.

techniques do not bring harm while capturing additional target-sensitive signals.

5. Micro-F1/accuracy is greatly affected by the majority class, as we can see the scores from Pos. and Micro are very consistent. TMNs, in fact, effectively improve the minority classes, which are reflected in Neg. and Neu., for example, JCI improves BL-MN by 3.78% in Neg. on Restaurant. This indicates their usefulness of capturing fine-grained sentiment signals. We will give qualitative examples in next section to show their modeling superiority for identifying target-sensitive sentiments.

Restaurant					
Model	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**
TRMN	69.00	68.66	50.66	87.70	78.86
RMN	67.48	66.48	49.11	86.85	77.14
Laptop					
Model	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**
TRMN	68.18	62.63	57.37	84.30	72.92
RMN	67.17	62.65	55.31	83.55	72.07

Table 4: Results with Recurrent Attention

Integration with Improved Attention: As discussed, the goal of this work is not for learning better attention but addressing the target-sensitive sentiment. In fact, solely improving attention does not solve our problem (see Sections 1 and 3). However, better attention can certainly help achieve an overall better performance for the ASC task, as it makes the targeted-context detection more accurate. Here we integrate our proposed technique JCI with a state-of-the-art sophisticated attention mechanism, namely, the recurrent attention framework, which involves multiple attentions learned iteratively (Kumar et al., 2016; Chen et al., 2017). We name our model with this integration as Target-sensitive Recurrent-attention Memory Network (TRMN) and the basic memory network with the recurrent attention as Recurrent-attention Memory Network (RMN). Their results are given in Table 4. TRMN achieves significant performance gain with $p < 0.05$ in paired t-test.

6.4 Effect of TCS Interaction for Identifying Target-Sensitive Sentiment

We now give some real examples to show the effectiveness of modeling TCS interaction for identifying target-sensitive sentiments, by comparing a regular MN and a TMN. Specifically, BL-MN and JPI are used. Other MNs/TMNs have similar performances to BL-MN/JPI qualitatively, so we do not list all of them here. For BL-MN and JPI, their sentiment scores of a single context word i are calculated by $\alpha_i W c_i$ (from Eq. 3) and $\alpha_i W_J tanh(W_1 c_i) + \alpha_i W_J \langle d_i, d_t \rangle tanh(W_2 c_i)$ (from Eq. 9), each of which results in a 3-dimensional vector.

Illustrative Examples: Table 5 shows two records in Laptop. In record 1, to identify the sentiment of target *price* in the presented sentence, the sentiment interaction between the context word "higher" and the target word *price* is the key. The

Record 1				Record 2			
Sentence	Price was higher when purchased on MAC..			**Sentence**	(MacBook) Air has higher resolution..		
Target	*Price*	**Sentiment**	*Negative*	**Target**	*Resolution*	**Sentiment**	*Positive*
Result	**Sentiment Logits on context "higher"**			**Result**	**Sentiment Logits on context "higher"**		
TMN	Negative	Neutral	Positive	**TMN**	Negative	Neutral	Positive
	0.2663 (Correct)	-0.2604	-0.0282		-0.4729	-0.3949	**0.9041 (Correct)**
MN	Negative	Neutral	Positive	**MN**	Negative	Neutral	Positive
	0.3641 (Correct)	-0.3275	-0.0750		**0.2562 (Wrong)**	-0.2305	- 0.0528

Table 5: Sample Records and Model Comparison between MN and TMN

specific sentiment scores of the word "higher" towards negative, neutral and positive classes in both models are reported. We can see both models accurately assign the highest sentiment scores to the negative class. We also observe that in MN the negative score (0.3641) in the 3-dimension vector $\{0.3641, -0.3275, -0.0750\}$ calculated by $\alpha_i W c_i$ is greater than neutral (-0.3275) and positive (-0.0750) scores. Notice that α_i is always positive (ranging in $(0, 1)$), so it can be inferred that the first value in vector Wc_i is greater than the other two values. Here c_i denotes the vector representation of "higher" so we use c_{higher} to highlight it and we have $\{Wc_{higher}\}^{Negative} > \{Wc_{higher}\}^{Neutral/Positive}$ as an inference.

In record 2, the target is *resolution* and its sentiment is positive in the presented sentence. Although we have the same context word "higher", different from record 1, it requires a positive sentiment interaction with the current target. Looking at the results, we see TMN assigns the highest sentiment score of word "higher" to positive class correctly, whereas MN assigns it to negative class. This error is expected if we consider the above inference $\{Wc_{higher}\}^{Negative} > \{Wc_{higher}\}^{Neutral/Positive}$ in MN. The cause of this unavoidable error is that Wc_i is not conditioned on the target. In contrast, $W_J\langle d_i, \cdot d_t\rangle tanh(W_2 c_i)$ can change the sentiment polarity with the aspect vector d_t encoded. Other TMNs also achieve it (like $W_I\langle d_i, d_t\rangle c_i$ in JCI).

One may notice that the aspect information (v_t) is actually also considered in the form of $\alpha_i W c_i + W v_t$ in MNs and wonder whether $W v_t$ may help address the problem given different v_t. Let us assume it helps, which means in the above example an MN makes $W v_{resolution}$ favor the positive class and $W v_{price}$ favor the negative class. But then we will have trouble when the context word is "lower", where it requires $W v_{resolution}$ to favor the negative class and $W v_{price}$ to favor the positive class. This contradiction reflects the theoretical problem discussed in Section 3.

Other Examples: We also found other interesting target-sensitive sentiment expressions like "*large bill*" and "*large portion*", "*small tip*" and "*small portion*" from Restaurant. Notice that TMNs can also improve the neutral sentiment (see Table 3). For instance, TMN generates a sentiment score vector of the context "over" for target aspect *price*: {**0.1373**, 0.0066, -0.1433} (negative) and for target aspect *dinner*: {0.0496, **0.0591**, -0.1128} (neutral) accurately. But MN produces both negative scores {**0.0069**, 0.0025, -0.0090} (negative) and {**0.0078**, 0.0028, -0.0102} (negative) for the two different targets. The latter one in MN is incorrect.

7 Conclusion and Future Work

In this paper, we first introduced the target-sensitive sentiment problem in ASC. After that, we discussed the basic memory network for ASC and analyzed the reason why it is incapable of capturing such sentiment from a theoretical perspective. We then presented six techniques to construct target-sensitive memory networks. Finally, we reported the experimental results quantitatively and qualitatively to show their effectiveness.

Since ASC is a fine-grained and complex task, there are many other directions that can be further explored, like handling sentiment negation, better embedding for multi-word phrase, analyzing sentiment composition, and learning better attention. We believe all these can help improve the ASC task. The work presented in this paper lies in the direction of addressing target-sensitive sentiment, and we have demonstrated the usefulness of capturing this signal. We believe that there will be more effective solutions coming in the near future.

Acknowledgments

This work was partially supported by National Science Foundation (NSF) under grant nos. IIS-1407927 and IIS-1650900, and by Huawei Technologies Co. Ltd with a research gift.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 452–461.

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. In *IEEE Transactions on Multimedia*, pages 1875–1886. IEEE.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 793–801.

Li Deng, Dong Yu, et al. 2014. Deep learning: methods and applications. In *Foundations and Trends® in Signal Processing*, pages 197–387. Now Publishers, Inc.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *ACM International Conference on Web Search and Data Mining*, pages 231–240.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 49–54.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Empirical Methods in Natural Language Processing*, pages 1746–1751.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Bing Liu. 2012. Sentiment analysis and opinion mining. In *Synthesis lectures on human language technologies*. Morgan & Claypool Publishers.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *European Chapter of the Association for Computational Linguistics*, pages 572–577.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *ACM International Conference on World Wide Web*, pages 347–356.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing*, pages 1412–1421.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *International Joint Conference on Artificial Intelligence*, pages 4068–4074.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *RANLP*, pages 378–382.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 2509–2514.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Empirical Methods in Natural Language Processing*, pages 79–86.

Livia Polanyi and Annie Zaenen. 2006. Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications*, pages 1–10. Springer.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task4: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2014)*. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing*, pages 1631–1642.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Empirical Methods in Natural Language Processing*, pages 214–224.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Empirical Methods in Natural Language Processing*, pages 606–615.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *International Conference on Learning Representations*.

Yunfang Wu and Miaomiao Wen. 2010. Disambiguating dynamic sentiment ambiguous adjectives. In *International Conference on Computational Linguistics*, pages 1191–1199.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. In *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253. Wiley Online Library.