

IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis

Navonil Majumder[†], Soujanya Poria[‡], Alexander Gelbukh[†],
Md. Shad Akhtar^Φ, Erik Cambria[‡], Asif Ekbal^Φ

[†]Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico

[‡]School of Computer Science and Engineering, Nanyang Technological University, Singapore

^ΦComputer Science and Engineering, Indian Institute of Technology, Patna, India

navo@nlp.cic.ipn.mx, sporia@ntu.edu.sg, gelbukh@gelbukh.com,
shad.pcs15@iitp.ac.in, cambria@ntu.edu.sg, asif@iitp.ac.in

Abstract

Sentiment analysis has immense implications in modern businesses through user-feedback mining. Large product-based enterprises like Samsung and Apple make crucial business decisions based on the large quantity of user reviews and suggestions available in different e-commerce websites and social media platforms like Amazon and Facebook. Sentiment analysis caters to these needs by summarizing user sentiment behind a particular object. In this paper, we present a novel approach of incorporating the neighboring aspects related information into the sentiment classification of the target aspect using memory networks. Our method outperforms the state of the art by 1.6% on average in two distinct domains.

1 Introduction

Sentiment analysis plays a huge role in user-feedback extraction from different popular e-commerce websites like Amazon, eBay, etc. Large enterprises are not only interested in the overall user sentiment about a given product, but the sentiment behind the finer aspects of a product is also very important to them. Companies allocate their resources to research, development, and marketing based on these factors. Aspect-based sentiment analysis (ABSA) caters to these needs.

Users tend to express their opinion on different aspects of a given product. For example, the sentence “*Everything is so easy to use, Mac software is just so much simpler than Microsoft software.*” expresses sentiment behind three aspects: “*use*”, “*Mac software*”, and “*Microsoft software*” to be positive, positive, and negative respectively. This leads to two tasks to be solved: aspect extraction (Shu et al., 2017) and aspect sentiment polarity detection (Wang et al., 2016). In this paper, we

tackle the latter problem by modeling the relation among different aspects in a sentence.

Recent works on ABSA does not consider the neighboring aspects in a sentence during classification. For instance, in the sentence “*The menu is very limited - I think we counted 4 or 5 entries.*”, the sub-sentence “*I think ... entries*” does not reflect the true sentiment behind containing aspect “*entries*”, unless the other aspect “*menu*” is considered. Here, the negative sentiment of “*menu*” induces “*entries*” to have the same sentiment. We hypothesize that our architecture iteratively models the influence from the other aspects to generate accurate target aspect representation.

In sentences containing multiple aspects, the main challenge an Aspect-Based-Sentiment-Analysis (ABSA) classifier faces is to correctly connect an aspect to the corresponding sentiment-bearing phrase (typically adjective). Let us consider this sentence “*Coffee is a better deal than overpriced cosi sandwiches*”. Here, we find two aspects: “*coffee*” and “*cosi sandwiches*”. It is clear in this sentence that the sentiment of “*coffee*” is expressed by the sentimentally charged word “*better*”; on the other hand, “*overpriced*” carries the sentiment of “*cosi sandwiches*”. The aim of the ABSA classifier is to learn these connections between the aspects and their sentiment bearing phrases.

In this work, we argue that during sentiment prediction of an aspect (say “*coffee*” in this case), the knowledge of the existence and representation of the other aspects (“*cosi sandwiches*”) in the sentence is beneficial. The sentiment of an aspect in a sentence can influence the succeeding aspects due to the presence of conjunctions. In particular, for sentences containing conjunctions like *and*, *not only*, *also*, *but*, *however*, *though*, etc., aspects tend

to share their sentiments. In the sentence “*Food is usually very good, though I wonder about freshness of raw vegetables*”, the aspect “*raw vegetables*” does not have any trivial sentiment marker linked to it. However, the positive sentiment of “*food*”, due to the word “*good*”, and presence of conjunction “*though*” determines the sentiment of “*raw vegetables*” to be negative. Thus, aspects when arranged as a sequence, reveal high correlation and interplay of sentiments.

To model these scenarios, firstly, following Wang et al. (2016), we independently generate aspect-aware sentence representations for all the aspects using gated recurrent unit (GRU) (Chung et al., 2014) and attention mechanism (Luong et al., 2015). Then, we employ memory networks (Sukhbaatar et al., 2015) to repeatedly match the target aspect representation with the other aspects to generate more accurate representation of the target aspect. This refined representation is fed to a softmax classifier for final classification. We empirically show below that our method outperforms the current state of the art (Ma et al., 2017) by 1.6% on average in two distinct domains: restaurant and laptop.

The rest of the paper structured as follows: Section 2 discusses previous works; Section 3 delves into the method we present; Section 4 mentions the dataset, baselines, and experimental settings; Section 5 presents and analyzes the results; finally, Section 6 concludes this paper.

2 Related Works

Sentiment analysis is becoming increasingly important due to the rise of the need to process textual data in wikis, micro-blogs, and other social media platforms. Sentiment analysis requires solving several related NLP problems, like aspect extraction (Poria et al., 2016). Aspect based sentiment analysis (ABSA) is a key task of sentiment analysis which focuses on classifying sentiment of each aspect in the sentences.

In this paper, we focus on ABSA, which is a key task of sentiment analysis that aims to classify sentiment of each aspect individually in a sentence. In recent days, thanks to the increasing progress of deep neural network research (Young et al., 2018), novel frameworks have been proposed, achieving notable performance improvement in aspect-based sentiment analysis.

The common way of doing ABSA is feeding the

aspect-aware sentence representation to the neural network for classification. This was first proposed by Wang et al. (2016) where they appended aspect embeddings with the each word embeddings of the sentence to generate aspect-aware sentence representation. This representation was further fed to an attention layer followed by softmax for final classification.

More recently, Ma et al. (2017) proposed a model where both context and aspect representations interact with each other’s attention mechanism to generate the overall representation. Tay et al. (2017) proposed word-aspect associations using circular correlation as an improvement over Wang et al. (2016)’s work. Also, Li et al. (2018) used transformer networks for target-oriented sentiment classification.

ABSA has also been researched from a question-answering perspective where deep memory networks have played a major role (Tang et al., 2016b; Li et al., 2017). However, unlike our proposed method, none of these methods have tried to model the inter-aspect relations.

3 Method

In this section, we formalize the task and present our method.

3.1 Problem Definition

Input We are given a sentence $S = [w_1, w_2, \dots, w_L]$, where w_i are the words and L is the maximum number of words in a sentence. Also, the given aspect-terms for sentence S are A_1, A_2, \dots, A_M , where $A_i = [w_k, \dots, w_{k+m-1}]$, $1 \leq k \leq L$, $0 < m \leq L - k + 1$, and M is the maximum number of aspects in a sentence.

Output Sentiment polarity (1 for positive, 0 for negative, and 2 for neutral) for each aspect-term A_i .

3.2 Model

The primary distinction between our model and the literature is the consideration of the neighboring aspects in a sentence with the target aspect. We assume that our inter-aspect relation modeling (IARM) architecture¹ models the relation between the target aspect and surrounding aspects, while filtering out irrelevant information. Fig. 1 depicts our model.

¹Implementation available on <http://github.com/senticnet/IARM>

3.2.1 Overview

Our IARM model can be summarized with the following steps:

Input Representation We replace the words in the input sentences and aspect-terms with pre-trained Glove word embeddings (Pennington et al., 2014). For multi-worded aspect-terms, we take the mean of constituent word embeddings as aspect representation.

Aspect-Aware Sentence Representation Following Wang et al. (2016), all the words in a sentence are concatenated with the given aspect representation. These modified sequence of words are fed to a gated recurrent unit (GRU)² for context propagation, followed by an attention layer to obtain the aspect-aware sentence representation; we obtain for all the aspects in a sentence.

Inter-Aspect Dependency Modeling We employ memory network (Sukhbaatar et al., 2015) to model the dependency of the target aspect with the other aspects in the sentence. This is achieved through matching target-aspect-aware sentence representation with aspect-aware sentence representation of the other aspects. After a certain number of iterations of the memory network, we obtain a refined representation of the sentence that is relevant to the sentiment classification of the target aspect. Further, this representation is passed to a softmax layer for final classification. The following subsections discuss these steps in details.

3.2.2 Input Representation

The words (w_i) in the sentences are represented with 300 (D) dimensional Glove word embeddings (Pennington et al., 2014), resulting sentence $S \in \mathbb{R}^{L \times D}$.

Similarly, aspect terms are represented with word embeddings. Multi-worded aspect terms are averaged over the constituent words. This results aspect representation $a_i \in \mathbb{R}^D$ for i^{th} aspect term.

3.2.3 Aspect-Aware Sentence Representation

It would be fair to assume that not all the words in a sentence carry sentimental information of a particular aspect (e.g., stop words have no impact). This warrants a sentence representation that reflects the sentiment of the given aspect. To achieve

²LSTM (Hochreiter and Schmidhuber, 1997) yields similar performance, but requires training more parameters

this, we first concatenate aspect a_i to all the words in the sentence S :

$$S_{a_i} = [w_1 \oplus a_i, w_2 \oplus a_i, \dots, w_L \oplus a_i] \in \mathbb{R}^{L \times 2D}. \quad (1)$$

In order to propagate the context information within the sentence, we feed S_{a_i} to a Gated Recurrent Unit (GRU) with output size D_s (kindly refer to Table 1 for the value). We denote this GRU as GRU_s . GRU_s is described as follows:

$$z = \sigma(x_t U_s^z + s_{t-1} W_s^z), \quad (2)$$

$$r = \sigma(x_t U_s^r + s_{t-1} W_s^r), \quad (3)$$

$$h_t = \tanh(x_t U_s^h + (s_{t-1} * r) W_s^h), \quad (4)$$

$$s_t = (1 - z) * h_t + z * s_{t-1}, \quad (5)$$

where h_t and s_t are the hidden outputs and the cell states respectively at time t . We obtain $R_{a_i} = GRU_s(S_{a_i})$, where $R_{a_i} \in \mathbb{R}^{L \times D_s}$ and the GRU_s has the following parameters: $U_s^z \in \mathbb{R}^{2D \times D_s}$, $W_s^z \in \mathbb{R}^{D_s \times D_s}$, $U_s^r \in \mathbb{R}^{2D \times D_s}$, $W_s^r \in \mathbb{R}^{D_s \times D_s}$, $U_s^h \in \mathbb{R}^{2D \times D_s}$, $W_s^h \in \mathbb{R}^{D_s \times D_s}$.

To amplify the sentimentally relevant words to aspect a_i , we employ an attention layer to obtain the aspect-aware sentence representation (it is effectively a refined aspect representation) r_{a_i} :

$$z = R_{a_i} W_s + b_s, \quad (6)$$

$$\alpha = \text{softmax}(z), \quad (7)$$

$$r_{a_i} = \alpha^T R_{a_i}, \quad (8)$$

where $z = [z_1, z_2, \dots, z_L] \in \mathbb{R}^{L \times 1}$, $\text{softmax}(x) = [e^{x_1} / \sum_j e^{x_j}, e^{x_2} / \sum_j e^{x_j}, \dots]$, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_L] \in \mathbb{R}^{L \times 1}$, $r_{a_i} \in \mathbb{R}^{D_s}$, $W_s \in \mathbb{R}^{D_s \times 1}$, and b_s is a scalar.

3.2.4 Inter-Aspect Dependency Modeling

We feed $R = [r_{a_1}, r_{a_2}, \dots, r_{a_M}] \in \mathbb{R}^{M \times D_s}$ to a GRU (GRU_a) of size D_o (kindly refer to Table 1 for the value) to propagate aspect information among the aspect-aware sentence representations and obtain $Q = GRU_a(R)$, where $Q \in \mathbb{R}^{M \times D_o}$ and GRU_a has the following parameters: $U_a^z \in \mathbb{R}^{D_s \times D_o}$, $W_a^z \in \mathbb{R}^{D_o \times D_o}$, $U_a^r \in \mathbb{R}^{D_s \times D_o}$, $W_a^r \in \mathbb{R}^{D_o \times D_o}$, $U_a^h \in \mathbb{R}^{D_s \times D_o}$, $W_a^h \in \mathbb{R}^{D_o \times D_o}$. This partially helps to model the dependency among aspects in a sentence.

After this, in order to further inter-aspect dependency modeling, we employ memory networks (Sukhbaatar et al., 2015), where the target-aspect representation (target-aspect-aware sentiment representation) r_{at} is supplied as the query.

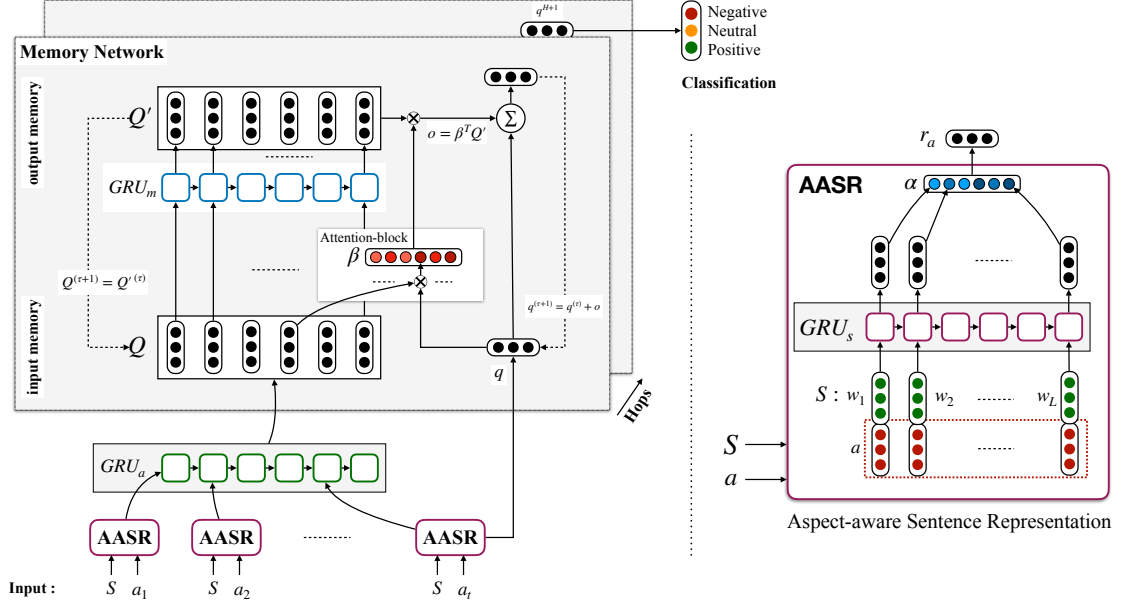


Figure 1: The proposed IARM architecture; AASR stands for *Aspect-Aware Sentence Representation*.

r_{a_t} is transformed into internal query state (q) with a fully connected layer as

$$q = \tanh(r_{a_t} W_T + b_T), \quad (9)$$

where $q \in \mathbb{R}^{D_o}$, $W_T \in \mathbb{R}^{D_s \times D_o}$, and $b_T \in \mathbb{R}^{D_o}$.

Input Memory Representation All the aspects in the sentence are stored in memory. Each aspect is represented by their corresponding aspect-aware sentence representation in Q . An attention mechanism is used to read these memories from Q (Weston et al., 2014). We compute the match between the query q and the memory slots in Q with inner product:

$$z = qQ^T, \quad (10)$$

$$\beta = \text{softmax}(z), \quad (11)$$

where $z = [z_1, z_2, \dots, z_M] \in \mathbb{R}^{M \times 1}$, $\beta = [\beta_1, \beta_2, \dots, \beta_M] \in \mathbb{R}^{M \times 1}$. Here, β_i is the measure of relatedness between target aspect and aspect i i.e., the attention score.

Output Memory Representation We choose the output memory vectors (Q') to be a refined version of the input memory vectors (Q), obtained by applying a GRU of size D_o (named GRU_m) on Q . Hence,

$$Q' = GRU_m(Q), \quad (12)$$

where GRU_m has the following parameters: $U_m^z \in \mathbb{R}^{D_o \times D_o}$, $W_m^z \in \mathbb{R}^{D_o \times D_o}$, $U_m^r \in \mathbb{R}^{D_o \times D_o}$, $W_m^r \in \mathbb{R}^{D_o \times D_o}$, $U_m^h \in \mathbb{R}^{D_o \times D_o}$, $W_m^h \in \mathbb{R}^{D_o \times D_o}$.

The response vector o is obtained by summing output vectors in Q' , weighted by the relatedness measures in β :

$$o = \beta^T Q', \quad (13)$$

where $o \in \mathbb{R}^{D_o}$.

Final Classification (Single Hop) In the case of single hop, target aspect representation q is added with memory output o to generate refined target aspect representative. This sum is passed to a softmax classifier of size C ($C = 3$ due to the classes of sentiment polarity):

$$\mathcal{P} = \text{softmax}((q + o)W_{smax} + b_{smax}), \quad (14)$$

$$\hat{y} = \underset{i}{\text{argmax}}(\mathcal{P}[i]), \quad (15)$$

where $W_{smax} \in \mathbb{R}^{D_o \times C}$, $b_{smax} \in \mathbb{R}^C$, and \hat{y} is the estimated sentiment polarity (0 for negative, 1 for positive, and 2 for neutral).

Multiple Hops We use total H (kindly refer to Table 1 for the value) number of hops in our model. Each hop generates a finer aspect representation q . Hence, we formulate the hops in the following way:

- Query (q) at the end of hop τ is updated as

$$q^{(\tau+1)} = q^{(\tau)} + o. \quad (16)$$

- Output memory vectors of hop τ , $Q^{(\tau)}$, is updated as the input memory vectors of hop $\tau + 1$:

$$Q^{(\tau+1)} = Q^{(\tau)}. \quad (17)$$

After H hops, $q^{(H)}$ becomes the target-aspect-aware sentence representation vector for the final classification:

$$\mathcal{P} = \text{softmax}(q^{(H+1)}W_{smax} + b_{smax}), \quad (18)$$

$$\hat{y} = \underset{i}{\text{argmax}}(\mathcal{P}[i]), \quad (19)$$

where $W_{smax} \in \mathbb{R}^{D_o \times C}$, $b_{smax} \in \mathbb{R}^C$, and \hat{y} is the estimated sentiment polarity (0 for negative, 1 for positive, and 2 for neutral). The whole algorithm is summarized in Algorithm 1.

3.3 Training

We train the network for 30 epochs using categorical cross entropy with L2-regularizer as loss function (L):

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{C-1} y_{ik} \log \mathcal{P}[k] + \lambda \|\theta\|_2, \quad (20)$$

where N is the number of samples, i is the sample index, k is the class value, λ is the regularization weight (we set it to 10^{-4}),

$$y_{ik} = \begin{cases} 1, & \text{if expected class value of sample } i \text{ is } k, \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

and θ is the set of parameters to be trained, where

$$\theta = \{U_{\{s,a,m\}}^z, W_{\{s,a,m\}}^z, U_{\{s,a,m\}}^r, W_{\{s,a,m\}}^r, U_{\{s,a,m\}}^h, W_{\{s,a,m\}}^h, W_s, b_s, W_T, b_T, W_{smax}, b_{smax}\}.$$

As optimization algorithm, Stochastic Gradient Descent (SGD)-based ADAM algorithm (Kingma and Ba, 2014) is used with learning-rate 0.001 due to its parameter-wise adaptive learning scheme.

Hyper-Parameters We employed grid-search to obtain the best hyper-parameter values. Table 1 shows the best choice of these values.

Algorithm 1 IARM algorithm

```

1: procedure TRAINANDTESTMODEL( $U, V$ )
   ▷  $U$  = train set,  $V$  = test set

2:   Aspect-aware sentence representation extraction:
3:   for  $i : [1, M]$  do ▷ generate for all the aspects in the sentence
4:      $r_{a_i} \leftarrow \text{AspectAwareSentRep}(S, a_i)$ 

5:   Query generation:
6:    $q \leftarrow \text{FCLayer}(r_{a_i})$  ▷ Transform the target-aspect-aware sentence representation to the query of memory network

7:   Memory networks:
8:    $Q \leftarrow \text{GRU}_a([r_{a_1}, r_{a_2}, \dots, r_{a_M}])$  ▷ initial input memory
9:    $Q' \leftarrow \text{GRU}_m(Q)$  ▷ initial output memory

10:  for  $i : [1, H]$  do ▷ memory network hops
11:     $z \leftarrow qQ^T$  ▷ match with target aspect
12:     $\beta \leftarrow \text{softmax}(z)$ 
13:     $o \leftarrow \beta^T Q'$  ▷ response vector
14:     $Q \leftarrow Q'$  ▷ input memory for the next hop
15:     $q \leftarrow q + o$  ▷ update target-aspect-aware sentence representation (query)

16:  Classification:
17:   $\hat{y} = \underset{j}{\text{argmax}}(\text{softmax}(q)[j])$  ▷ softmax classification
18:   $\text{TestModel}(V)$ 

19: procedure ASPECTAWARESENTREP( $S, a$ ) ▷ generation of aspect-aware sentence representation
20:   $R_a \leftarrow \text{GRU}_s([w_1 \oplus a, w_2 \oplus a, \dots, w_L \oplus a])$  ▷  $S = [w_1, w_2, \dots, w_L]$ 
21:   $z \leftarrow \text{FCLayer}(R_a)$ 
22:   $\alpha \leftarrow \text{softmax}(z)$ 
23:   $r_a \leftarrow \alpha^T R_a$ 
24:  return  $r_a$ 

25: procedure TESTMODEL( $V$ )
26:  Similar to the training phase,  $V$  is passed through the learnt models to get the classification outputs. Section 3.3 mentions the trainable parameters ( $\theta$ ).

```

4 Experiments

In this section, we discuss the dataset used and different experimental settings devised for the evaluation of our model.

4.1 Dataset Details

We evaluate our model with SemEval-2014 ABSA dataset³. It contains samples from two different domains: Restaurant and Laptop. Table 2 shows the distribution of these samples by class labels. Also, Table 3 shows the count of the samples with single aspect sentence and multi-aspect sentence.

³<http://alt.qcri.org/semeval2014/task4>

4.2 Baseline Methods

We compare our method against the following baseline methods:

LSTM Following Wang et al. (2016), the sentence is fed to a long short-term memory (LSTM) network to propagate context among the constituent words. The mean of all the hidden outputs from the LSTM is taken as the sentence representation, which is fed to a softmax classifier. Aspect-terms have no participation in the classification process.

TD-LSTM Following Tang et al. (2016a), sequence of words preceding (left context) and succeeding (right context) target aspect term are fed to two different LSTMs. Mean of the hidden outputs of the LSTMs are concatenated and fed to softmax classifier.

AE-LSTM Following Wang et al. (2016), the sentence is fed to an LSTM for context propagation. Then, the hidden outputs are concatenated with target-aspect representation, from which attention scores are calculated. Hidden outputs are pooled based on the attention scores to generate intermediate aspect representation. Final representation is generated as the sum of the affine transformations of intermediate representation and final LSTM hidden output. This representation is fed to softmax classifier.

ATAE-LSTM Following Wang et al. (2016), ATAE-LSTM is identical to AE-LSTM, except the LSTM is fed with the concatenation of aspect-term representation and word representation.

IAN Following Ma et al. (2017), target-aspect and its context are sent to two distinct LSTMs and the means of the hidden outputs are taken as intermediate aspect representation and context representation respectively. Attention scores are generated from the hidden outputs of both LSTMs which is used to generate final aspect and context representation. The concatenation of these two vectors are sent to a softmax classifier for final classification.

| Hyper-Parameter | Restaurant | Laptop |
|-----------------|------------|--------|
| D_s | 300 | 400 |
| D_o | 350 | 400 |
| Hop Count | 3 | 10 |

Table 1: Hyper-parameter choices.

| Domain | Positive | | Negative | | Neutral | |
|------------|----------|------|----------|------|---------|------|
| | Train | Test | Train | Test | Train | Test |
| Restaurant | 2164 | 728 | 805 | 196 | 633 | 196 |
| Laptop | 987 | 341 | 866 | 128 | 460 | 169 |

Table 2: Distribution of the samples by class labels in SemEval 2014 dataset.

4.3 Experimental Settings

In order to draw a comprehensive comparison between our IARM model and the baseline methods, we performed the following experiments:

Overall Comparison IARM is compared with the baseline methods for both of the domains.

Single Aspect and Multi Aspect Scenarios In this setup, samples with single aspect and multi aspect sentences are tested independently on the trained model. For IAN, we ran our own experiments for this scenario.

Cross-Domain Evaluation Here, the model trained for restaurant domain is tested with the test set for laptop domain and vice versa. For IAN, we ran our own experiments for this scenario.

5 Results and Discussion

We discuss the results of different experiments below:

Overall Comparison We present the overall performance of our model against the baseline methods in Table 4.

It is evident from the results that our IARM model outperforms all the baseline models, including the state of the art, in both of the domains. We obtained bigger improvement in laptop domain, of 1.7%, compared to restaurant domain, of 1.4%. This shows that the inclusion of the neighboring aspect information and memory network has an overall positive impact on the classification process.

Single Aspect and Multi-Aspect Scenarios Following Table 5, our IARM model beats the

| Domain | Train | | Test | |
|------------|-------|------|------|-----|
| | SA | MA | SA | MA |
| Restaurant | 1007 | 2595 | 285 | 835 |
| Laptop | 917 | 1396 | 259 | 379 |

Table 3: Distribution of the samples by single aspect/multi aspect sentence criteria in SemEval 2014 (SA: Single Aspect, MA: Multi Aspect).

| Model | Domain | |
|-----------------|-------------|-------------|
| | Restaurant | Laptop |
| Majority Voting | 53.4 | 65.0 |
| LSTM | 74.3 | 66.5 |
| TD-LSTM | 75.6 | 68.1 |
| AE-LSTM | 76.2 | 68.9 |
| ATAE-LSTM | 77.2 | 68.7 |
| IAN (SoA) | 78.6 | 72.1 |
| IARM | 80.0 | 73.8 |

Table 4: Domain-wise accuracy (%) of the discussed models. Best accuracy for each domain is marked with bold font.

state of the art in both single aspect and multi-aspect scenarios in both of the domains. It is interesting that both model perform better in multi-aspect scenario for restaurant domain. However, for laptop domain IAN performs better in single aspect scenario, even though there are more multi-aspect samples than single aspect samples (shown in Table 3). This indicates the failure of IAN model to learn multi-aspect scenario, where IARM model performs significantly better.

| Model | Restaurant | | Laptop | |
|-----------|-------------|--------------|-------------|-------------|
| | SA | MA | SA | MA |
| IAN (SoA) | 75.4 | 77.7 | 72.5 | 71.6 |
| IARM | 78.6 | 80.48 | 73.4 | 74.1 |

Table 5: Accuracy of the models for single aspect and multi aspect scenario; SA: Single Aspect, MA: Multi Aspect.

Cross-Domain Evaluation Following Table 6, IARM outperforms the state of the art IAN by 2% in both cross-domain scenarios. This indicates the ability of IARM in learning general domain-independent semantic structures from the training data.

| Model | Rest \rightarrow Lap | Lap \rightarrow Rest |
|-----------|------------------------|------------------------|
| IAN (SoA) | 64.6 | 72.0 |
| IARM | 66.7 | 74.0 |

Table 6: Accuracy for cross-domain evaluation; Rest: Restaurant domain, Lap: Laptop domain; A \rightarrow B signifies train-set is the train-set of domain A and test-set is the test-set of domain B.

5.1 Case Study

We analyze and compare IARM and IAN with single aspect and multi-aspect samples from the SemEval 2014 dataset.

Single Aspect Case It is evident from Table 5, that IARM outperforms IAN in single-aspect scenario. For example, the sentence “*I recommend any of their salmon dishes.....*” having aspect “*salmon dishes*”, with positive sentiment, fails to be correctly classified by IAN as the attention network focuses on the incorrect word “*salmon*”, as shown in Fig. 2a. Since, “*salmon*” does not carry any sentimental charge, the network generates an ineffective aspect-aware sentiment representation, which leads to misclassification.

On the other hand, IARM succeeds in this case, because the word-level attention network generates correct attention value as α in Eq. (7). α for this case is depicted in Fig. 2b, where it is clear that the network emphasizes the correct sentiment-bearing word “*recommended*”. This leads to effective aspect-aware sentence representation by the network, making correct final classification.

Multi-Aspect Case IARM also outperforms IAN in multi-aspect scenario, which can be observed in Table 5. We suspect that the presence of multiple aspects in sentence makes IAN network perplexed as to the connection between aspect and the corresponding sentiment-bearing word in the sentence. For example, the sentence “*Coffee is a better deal than overpriced cosi sandwiches*” contains two aspects: “*coffee*” and “*better*”. Clearly, the sentiment behind aspect “*coffee*” comes from the word “*better*” and the same for aspect “*cosi sandwiches*” comes from “*overpriced*”. However, IAN fails to make this association for the aspect “*cosi sandwiches*”, evident from the attention weights of IAN shown in Fig. 3a where the emphasis is on “*better*”. This leads to imperfect aspect-aware sentence representation generation, resulting misclassification of the target aspect to be positive.

However, IARM resolves this issue with the combination of word-level aspect aware attention (α) and the memory network. Since, the memory network compares the target-aspect-aware sentence representation with the sentence representations for the other aspects repeatedly, eventually the correct representation for the target aspect emerges from the memory network.

Also, the consideration of surrounding aspects forces the network to better distinguish the sentiment-bearing words for a particular aspect. These points are reflected in the α attention weights of the aspects “*coffee*” and “*cosi sand-*



Figure 2: Attention weights for IAN and IARM for “*I recommend any of their salmon dishes*”.

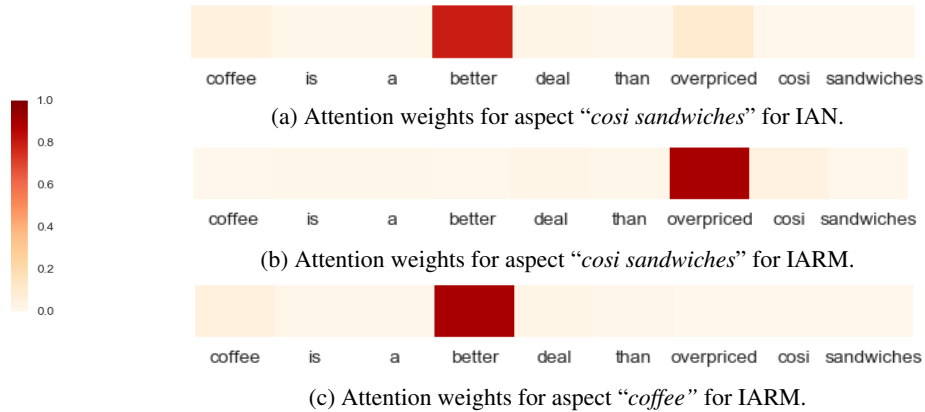


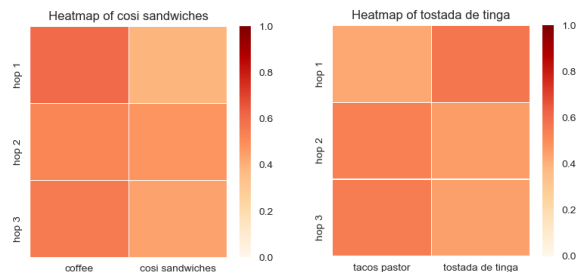
Figure 3: Attention weights for IAN and IARM for the sentence “*Coffee is a better deal than overpriced cosi sandwiches*”.

wiches”, shown in Fig. 3b and Fig. 3c respectively, where the network emphasizes the correct sentiment-bearing words for each aspect, “*better*” and “*overpriced*”, respectively. Again, the memory network compares the target aspect-aware sentence representation for “*cosi sandwiches*” with the same for “*coffee*” and incorporates relevant information into the target-aspect representation q in Eq. (16) along several hops.

This phenomenon is indicated in Fig. 4a, where the degree of incorporation of the aspect terms is measured by the attention weights β in Eq. (11). Here, the network is incorporating information from aspect “*coffee*” into aspect “*cosi sandwiches*” over three hops. We surmise that this information is related to the sentiment-bearing word “*better*” of the aspect “*coffee*”, because a comparison using the word “*better*” implies the presence of a good (“*coffee*”) and a bad (“*cosi sandwiches*”) object. However, this semantics is misconstrued by IAN, which leads to aspect misclassification.

IARM performs considerably well when conjunction plays a vital role in understanding the sentence structure and meaning for sentiment analysis. For example, “*my favs here are the tacos pastor and the tostada de tinga*” where the aspects

“*tacos pastor*” and “*tostada de tinga*” are connected using conjunction “*and*” and both rely on the sentiment bearing word *favs*. Such complex relation between the aspects and the corresponding sentiment-bearing word is grasped by IARM as shown in Fig. 4b. Another example where the inter-aspect relation is necessary for the correct classification is shown in Fig. 5, where the aspects “*atmosphere*” and “*service*” both rely on the sentiment bearing word “*good*”, due to the conjunction “*and*”.



(a) Memory network attention weights for the sentence “*Coffee is a better deal than overpriced cosi sandwiches*”.
 (b) Memory network attention weights for the sentence “*my favs here are the tacos pastor and the tostada de tinga*”.

Figure 4: Memory network attention weights for IARM.

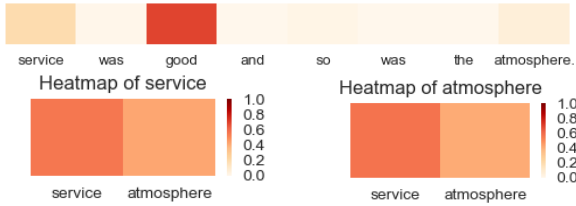


Figure 5: Memory network attention weights for IARM for the sentence “*service was good and so was the atmosphere.*” The word importance heatmap is for the aspect “*atmosphere*”.

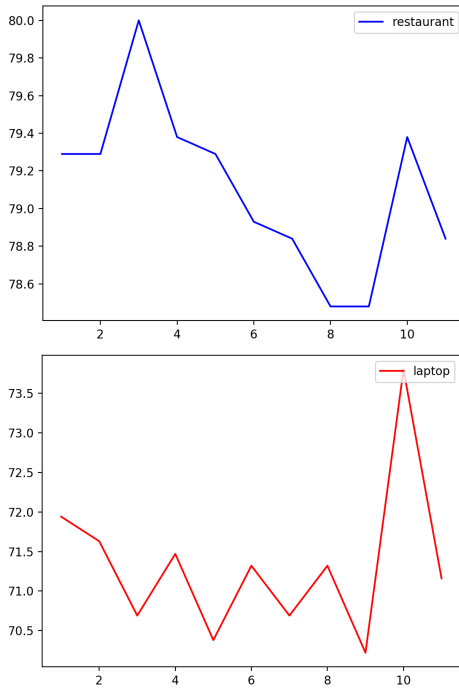


Figure 6: Hop-Accuracy plot for both domains.

5.2 Error Analysis

IARM also fails to correctly classify in some cases, e.g., in the sentence “*They bring a sauce cart up to your table and offer you 7 or 8 choices of sauces for your steak (I tried them ALL).*”, the aspect “*choices of sauces*” is misclassified by our network as neutral. This happened due to the IARM’s inability to correctly interpret the positive sentiment behind “*7 or 8 choices of sauces*”.

Again, the IARM could not correctly classify aspect the “*bread*s” to be positive in the sentence “*Try the homemade breads.*”. This happened, because the word “*try*” itself is not sentimentally charged, but can carry sentimental meaning given the right context. This context was not recognized by IARM, which led to misclassification.

5.3 Hop-Performance Relation

In our experiments, we tried different hop counts of the memory network. We observed that the network performs best with three hops for restaurant domain and ten hops for laptop domain, which is shown in the hop count - performance plot in Figure Fig. 6. It can be observed that the plot for restaurant domain is smoother than the plot for laptop domain. We assume that this is due to the restaurant domain having higher number of samples than laptop domain, as shown in Table 2.

Also, the plot for restaurant domain shows a downward trend over the increasing number of hops, with spikes in hop 3, hop 10. This suggests a irregular cyclic nature of the memory network where those certain hop counts yields higher quality representations than their neighbor. The same cannot be said for laptop domain as the plot presents a zig-zag pattern.

6 Conclusion

In this paper, we presented a new framework, termed IARM, for aspect-based sentiment analysis. IARM leverages recurrent memory networks with multihop attention mechanism. We empirically illustrate that an aspect in a sentence is influenced by its neighboring aspects. We exploit this property to obtain state-of-the-art performance in aspect-based sentiment analysis in two distinct domains: restaurant and laptop. However, there remains plenty of room for improvement in the memory network, e.g., for generation of better aspect-aware representations.

References

- Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.
- Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deep memory networks for attitude identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 671–680. ACM.

- Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956, Melbourne, Australia. Association for Computational Linguistics.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4068–4074.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.
- Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. *arXiv preprint arXiv:1705.00251*.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS’15*, pages 2440–2448, Cambridge, MA, USA. MIT Press.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. 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. The COLING 2016 Organizing Committee.
- Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*.
- Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. *arXiv preprint arXiv:1712.05403*.
- Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. 2016. 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. Association for Computational Linguistics.
- Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.
- Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.