Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension

Yichun Yin[1], **Yangqiu Song**[2], **Ming Zhang**[1]
[1]School of Electronics Engineering and Computer Science, Peking University, Beijing, China
[2]Department of Computer Science and Engineering, HKUST, Hong Kong
{yichunyin, mzhang_cs}@pku.edu.cn, yqsong@cse.ust.hk

Abstract

Document-level multi-aspect sentiment classification is an important task for customer relation management. In this paper, we model the task as a machine comprehension problem where pseudo question-answer pairs are constructed by a small number of aspect-related keywords and aspect ratings. A hierarchical iterative attention model is introduced to build aspect-specific representations by frequent and repeated interactions between documents and aspect questions. We adopt a hierarchical architecture to represent both word level and sentence level information, and use the attention operations for aspect questions and documents alternatively with the multiple hop mechanism. Experimental results on the TripAdvisor and BeerAdvocate datasets show that our model outperforms classical baselines.

1 Introduction

Document-level sentiment classification is one of the pragmatical sentiment analysis tasks (Pang and Lee, 2007; Liu, 2010). There are many Web sites having platforms for users to input reviews over products or services, such as TripAdvisor, Yelp, Amazon, etc. Most of reviews are very comprehensive and thus long documents. Analyzing these documents to predict ratings of products or services is an important complementary way for better customer relationship management. Recently, neural network based approaches have been developed and become state-of-the-arts for long-document sentiment classification (Tang et al., 2015a,b; Yang et al., 2016). However, predicting an overall score for each long document is not enough, because the document can mention different aspects of the corresponding product or service. For example, in Figure 1, there could be different aspects for a review of hotel. These aspects help customer service better understand what are the major pros and cons of the product or service. Compared to the overall rating, users are less motivated to give aspect ratings. Therefore, it is more practically useful to perform document-level multi-aspect sentiment classification task, predicting different ratings for each aspect rather than an overall rating.

Figure 1: Example: hotel review with aspects.

One straightforward approach for document-level multi-aspect sentiment classification is multi-task learning (Caruana, 1997). For neural networks, we can simply treat each aspect (e.g., rating from one to five) as a classification task, and let different tasks use softmax classifier to extract task-specific representations at the top layer while share the input and hidden layers to mutually enhance the prediction results (Collobert et al., 2011; Luong et al., 2016). However, such approach ignores the fact that the aspects themselves have semantic meanings. For example, as human beings, if we were asked to evaluate the aspect rating of a document, we simply read the review, and find aspect-related keywords, and see around comments. Then, we aggregate all the related snippets to make a decision.

In this paper, we propose a novel approach to treat document-level multi-aspect sentiment clas-

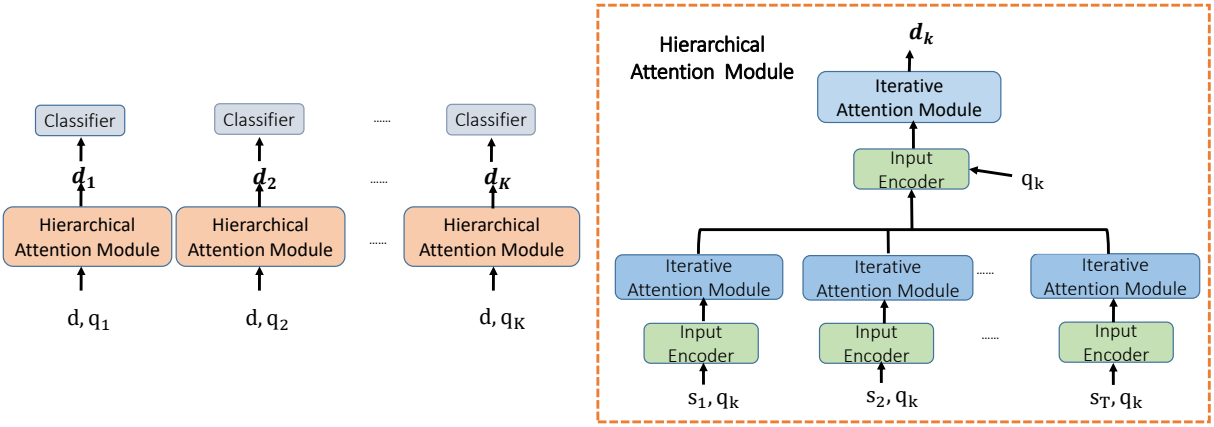

Figure 2: The architecture of our model. Left: multi-task learning. Right: hierarchical attention module which includes input encoders and iterative attention modules.

sification as a machine comprehension (Kumar et al., 2016; Sordoni et al., 2016) problem. To mimic human's evaluation of aspect classification, we create a list of keywords for each aspect. For example, when we work on the *Room* aspect, we generate some keywords such as "room," "bed," "view," etc. Then we can ask pseudo questions: "How is the room?" "How is the bed?" "How is the view?" and provide an answer "Rating 5." In this case, we can train a machine comprehension model to automatically attend corresponding text snippets in the review document to predict the aspect rating. Specifically, we introduce a hierarchical and iterative attention model to construct aspect-specific representations. We use a hierarchical architecture to build up different representations at both word and sentence levels interacting with aspect questions. At each level, the model consists of input encoders and iterative attention modules. The input encoder learns memories[1] of documents and questions with Bi-directional LSTM (Bi-LSTM) model and non-linear mapping respectively. The iterative attention module takes into memories as input and attends them sequentially with a multiple hop mechanism, performing effective interactions between documents and aspect questions.

To evaluate the effectiveness of the proposed model, we conduct extensive experiments on the TripAdvisor and BeerAdvocate datasets and the results show that our model outperforms typical baselines. We also analyze the effects of numbers of the hop and aspect words on performances. Moreover, a case study for attention results is performed at both word and sentence levels.

The contributions of this paper are two-fold. First, we study the document-level multi-aspect sentiment classification as a machine comprehension problem and introduce a hierarchical iterative attention model for it. Second, we demonstrate the effectiveness of proposed model on two datasets, showing that our model outperforms classical baselines. The code and data for this paper are available at `https://github.com/HKUST-KnowComp/DMSCMC`.

2 Method

In this section, we introduce our proposed method.

2.1 Problem Definition and Hierarchical Framework

We first briefly introduce the problem we work on. Given a piece of review, our task is to predict the ratings of different aspects. For example, in Figure 1, we predict the ratings of *Cleanliness*, *Room*, and *Value*. To achieve this, we assume that there are existing reviews with aspect ratings for machines to learn. Formally, we denote the review document as d containing a set of T_d sentences $\{s_1, s_2, \ldots s_{T_d}\}$. For the t-th sentence s_t, we use a set of words $\{w_1, w_2, \ldots w_{|s_t|}\}$ to represent it, and use $\mathbf{w}_i$, $\mathbf{w}_i^w$ and $\mathbf{w}_i^p$ as the one-hot encoding, word embedding, and phrase embedding for w_i respectively. The phrase embedding encodes the semantics of phrases where the current word w_i is the center (e.g., hidden vectors learned by Bi-LSTM shown in Section 2.2). For each q_k of K aspects

[1]Following the work (Weston et al., 2015; Sukhbaatar et al., 2015), we refer the memory to a set vectors which are stacked together and could be attended.

$\{q_1, q_2, \ldots, q_K\}$, we use N_k aspect-related keywords, $\left\{q_{k_1}, q_{k_2} \ldots q_{k_{N_k}}\right\}$, to represent it. Similarly, we use $\mathbf{q}_{k_i}$, $\mathbf{q}^w_{k_i}$ as the one-hot encoding and word embedding for q_{k_i} respectively.

There are several sophisticated methods for choosing aspect keywords (e.g., topic model). Here, we consider a simple way where five seeds were first manually selected for each aspect and then more words were obtained based on their cosine similarities with seeds[2]

As shown in Figure 2 (left), our framework follows the idea of multi-task learning, which learns different aspects simultaneously. In this case, all these tasks share the representations of words and architecture of semantic model for the final classifiers. Different from straightforward neural network based multi-task learning (Collobert et al., 2011), for each document d and an aspect q_k, our model uses both the content of d and all the related keywords $\left\{q_{k_1}, q_{k_2} \ldots q_{k_{N_k}}\right\}$ as input. Since the keywords can cover most of the semantic meanings of the aspect, and we do not know which document mentions which semantic meaning, we build an attention model to automatically decide it (introduced in Section 2.3). Assuming that the keywords have been decided, we use a hierarchical attention model to select useful information from the review documents. As shown in Figure 2 (right), the hierarchical attention of keywords is applied to both sentence level (to select meaningful words) and document level (to select meaningful sentence). Thus, our model builds aspect-specific representations in a bottom-up manner.

Specifically, we obtain sentence representations $\left\{\mathbf{s}^k_1, \mathbf{s}^k_2, \ldots \mathbf{s}^k_T\right\}$ using the input encoder (Section 2.2) and iterative attention module (Section 2.3) at the word level. Then we take sentence representations and k-th aspect as input and apply the sentence-level input encoder and attention model to generate the document representation $\mathbf{d}_k$ for final classification. As shown in Figure 2 (right), the attention model is applied twice at different levels of the representation.

2.2 Input Encoder

The input module builds memory vectors for the iterative attention module and is performed both at word and sentence levels. For a document, it converts word sequence into word level memory $\mathbf{M}^d_w$ and sentence sequence into sentence level memory $\mathbf{M}^d_s$ respectively. For an aspect question q_k, it takes a set of aspect-specific words $\{q_{k_i}\}_{1 \le i \le N_k}$ as input and derives word level memory $\mathbf{M}^q_w$ and sentence level memory $\mathbf{M}^q_s$.

To construct $\mathbf{M}^d_w$, we obtain word embeddings $\left\{\mathbf{w}^w_1, \mathbf{w}^w_2, \ldots \mathbf{w}^w_{|s_t|}\right\}$ from an embedding matrix $\mathbf{E}^A$ applied to all words shown in the corpus. Then, LSTM (Hochreiter and Schmidhuber, 1997) model is used as the encoder to produce hidden vectors of words based on the word embeddings. At each step, LSTM takes input $\mathbf{w}^w_t$ and derives a new hidden vector by $\mathbf{h}_t = \text{LSTM}(\mathbf{w}^w_t, \mathbf{h}_{t-1})$. To preserve the subsequent context information for words, another LSTM is ran over word sequence in a reverse order simultaneously. Then the forward hidden vector $\overrightarrow{\mathbf{h}}_t$ and backward hidden vector $\overleftarrow{\mathbf{h}}_t$ are concatenated as phrase embedding $\mathbf{w}^p_t$. We stack these phrase embeddings together as word level memory $\mathbf{M}^d_w$. Similarly, we feed sentence representations into another Bi-LSTM to derive the sentence level memory $\mathbf{M}^d_s$. Note that, the sentence representations are obtained using the iterative attention module which is described as Eq. (5) in Section 2.3.

Since we have question keywords as input, to allow the interactions between questions and documents, we also build question memory in following way. We obtain $\mathbf{Q}_k = \left\{\mathbf{q}^w_{k_i}\right\}_{1 \le i \le N_k}$ by looking up an embedding matrix [3] $\mathbf{E}^B$ applied to all question keywords. Then a non-linear mapping is applied to obtain the question memory at word level:

$$\mathbf{M}^{q_k}_w = \tanh(\mathbf{Q}_k \mathbf{W}^q_w), \quad (1)$$

where $\mathbf{W}^q_w$ is the parameter matrix to adapt q_k at word level. Similarly, we use another mapping to obtain the sentence level memory:

$$\mathbf{M}^{q_k}_s = \tanh(\mathbf{Q}_k \mathbf{W}^q_s), \quad (2)$$

where $\mathbf{W}^q_s$ is the parameter matrix to adapt q_k at sentence level.

2.3 Iterative Attention Module

The iterative attention module (IAM) attends and reads memories of questions and documents alternatively with a multi-hop mechanism, deriving

[2] For example, the words "value," "price," "worth," "cost," and "$" are selected as seeds for aspect *Price*. The information for seeds can be found in our released resource.

[3] $\mathbf{E}^A$ and $\mathbf{E}^B$ are initialized by the same pre-trained embeddings but are different embedding matrices with different updates.

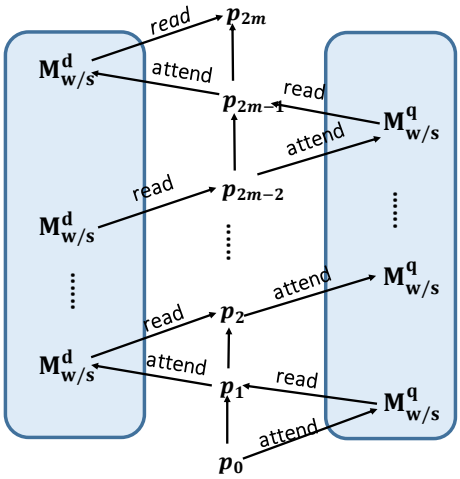

Figure 3: The iterative attention module.

aspect-specific sentence and document representations. As we discussed in the introduction, the set of selected question keywords may not best characterize the aspect for different documents. Thus, the IAM module introduces a backward attention to use document information (word or sentence) to select useful keywords of each aspect as the document-specific question to build attention model.

The illustration of IAM is shown in Figure 3. To obtain sentence representations, it takes $\mathbf{M}^d_w$ and $\mathbf{M}^q_w$ as the input and performs m iterations (hops). For each iteration, IAM conducts four operations: (1) attends the question memory by the selective vector $\mathbf{p}$ and summarizes question memory vectors into a single vector $\hat{\mathbf{q}}$; (2) updates the selective vector by the previous one and $\hat{\mathbf{q}}$; (3) attends document (content) memory based on the updated selective vector and summarizes memory vectors in to a single vector $\hat{\mathbf{c}}$; (4) updates the selective vector by the previous one and $\hat{\mathbf{c}}$.

We unify operations (1) and (3) by an attention function $\hat{\mathbf{x}} = \mathcal{A}(\mathbf{p}, \mathbf{M})$, where $\mathbf{M}$ could be $\mathbf{M}^d_w$ or $\mathbf{M}^q_w$ which corresponds $\hat{\mathbf{x}} = \hat{\mathbf{c}}$ or $\hat{\mathbf{x}} = \hat{\mathbf{q}}$. The attention function $\mathcal{A}$ is decomposed as:

$$\begin{aligned} \mathbf{H} &= \tanh(\mathbf{M}\mathbf{W}_a \odot (\mathbb{1}\mathbf{p})) \\ \mathbf{a} &= \text{softmax}(\mathbf{H}\mathbf{v}_a^T) \\ \hat{\mathbf{x}} &= \sum \mathbf{a}_i \mathbf{M}_i, \end{aligned} \tag{3}$$

where $\mathbb{1}$ is a vector with all elements are 1, which copies the selective vector to meet the dimension requirement. The $\mathbf{W}_a$ and $\mathbf{v}_a$ are parameters, $\mathbf{a}$ is attention weights for memory vectors, and $\mathbf{M}_i$ means i-th row in $\mathbf{M}$.

Operations (2) and (4) are formulated as an update function $\mathbf{p}_{2i-\{l\}} = \mathcal{U}(\hat{\mathbf{x}}, \mathbf{p}_{2i-\{l\}-1})$, where i is the hop index, l can be 0 or 1 which corresponds to $\hat{\mathbf{x}} = \hat{\mathbf{c}}$ or $\hat{\mathbf{x}} = \hat{\mathbf{q}}$ respectively. We initialize $\mathbf{p}_0$ by a zero vector. The update function $\mathcal{U}$ can be a recurrent neural network (Xiong et al., 2017) or other heuristic weighting functions. In this paper, we introduce a simple strategy:

$$\mathbf{p}_{2i-\{l\}} = \hat{\mathbf{x}}, \tag{4}$$

which ignores the previous selective vector but succeeds to obtain comparable results with other more complicated function in the initial experiments.

Multi-hop mechanism attends different memory locations in different hops (Sukhbaatar et al., 2015), capturing different interactions between documents and questions. In order to preserve the information of various kinds of interactions, we concatenate all $\hat{\mathbf{c}}$'s in each hop as the final representations of sentences:

$$\mathbf{s} = [\hat{\mathbf{c}}_1; \hat{\mathbf{c}}_2; \cdots \hat{\mathbf{c}}_m]. \tag{5}$$

After obtaining sentence representations, we feed them into the sentence-level input encoder, deriving the memories $\mathbf{M}^d_s$ and $\mathbf{M}^q_s$. Then, the aspect-specific document representation $\mathbf{d}_k$ is obtained by the sentence-level IAM in a similar way.

2.4 Objective Function

For each aspect, we obtain aspect-specific document representations $\{\mathbf{d}_k\}_{1 \le k \le K}$. All these representations are fed into classifiers, each of which includes a softmax layer. The softmax layer outputs the probability distribution over $|\mathcal{Y}|$ categories for the distributed representation, which is defined as:

$$\mathbf{p}'(d,k) = \text{softmax}(\mathbf{W}^{class}_k \mathbf{d}_k), \tag{6}$$

where $\mathbf{W}^{class}_k$ is the parameter matrix.

We define the cross-entropy objective function between gold sentiment distribution $\mathbf{p}(d,k)$ and predicted sentiment distribution $\mathbf{p}'(d,k)$ as the classification loss function:

$$-\sum_{d \in \mathcal{D}} \sum_{k=1}^{K} \sum_{i=1}^{|\mathcal{Y}|} \mathbf{p}(d,k) \log(\mathbf{p}'(d,k)), \tag{7}$$

where $\mathbf{p}(d,k)$ is a one-hot vector, which has the same dimension as the number of classes, and only the dimension associated with the ground truth label is one, with others being zeros.

Dataset	#docs	#words/doc	#words/sent
TripAdvisor	29,391	251.7	18.0
BeerAdvocate	51,020	144.5	12.1

Table 1: Statistics of the datasets. The rating scale of TripAdvisor dataset is 1-5. The rating scale of BeerAdvocate dataset is 1-10.

3 Experiment

In this section, we show experimental results to demonstrate our proposed algorithm.

3.1 Datasets

We conduct our experiments on TripAdvisor (Wang et al., 2010) and BeerAdvocate (McAuley et al., 2012; Lei et al., 2016) datasets, which contain seven aspects (*value*, *room*, *location*, *cleanliness*, *check in/front desk*, *service*, and *business service*) and four aspects (*feel*, *look*, *smell*, and *taste*) respectively. We follow the processing step (Lei et al., 2016) by choosing the reviews with different aspect ratings and the new datasets are described in Table 1. We tokenize the datasets by Stanford corenlp[4] and randomly split them into training, development, and testing sets with 80/10/10%.

3.2 Baseline Methods

To demonstrate the effectiveness of the proposed method, we compare our model with following baselines:

Majority uses the majority sentiment label in development sets as the predicted label.

SVM uses unigram and bigram as text features and uses Liblinear (Fan et al., 2008) for learning.

SLDA refers to supervised latent Dirichlet allocation (Blei and Mcauliffe, 2010) which is a statistical model of labeled documents.

NBoW is a neural bag-of-words model averaging embeddings of all words in a document and feeds the resulted embeddings into SVM classifier.

DAN is a deep averaging network model which consists of several fully connected layers with averaged word embeddings as input. One novel word dropout strategy is employed to boost model performances (Iyyer et al., 2015).

CNN continuously performs a convolution operation over a sentence to extract words neighboring features, then gets a fixed-sized representation by a pooling layer (Kim, 2014).

[4]http://nlp.stanford.edu/software/corenlp.shtml

LSTM is one variant of recurrent neural network and has been proved to be one of state-of-the-art models for document-level sentiment classification (Tang et al., 2015a). We use LSTM to refer Bi-LSTM which captures both forward and backward semantic information.

HAN means the hierarchical attention network which is proposed in (Yang et al., 2016) for document classification. Note that, the original *HAN* depends GRU as the encoder. In our experiments, LSTM-based HAN obtains slightly better results. Thus, we report the results of HAN with LSTM as the encoder.

We extend *DAN*, *CNN*, *LSTM* with the hierarchical architecture and multi-task framework, the corresponding models are *MHDAN*, *MHCNN* and *MHLSTM* respectively. Besides, *MHAN* is also evaluated as one baseline, which is *HAN* with the multi-task learning.

3.3 Implementation Details

We implement all neural models using Theano (Theano Development Team, 2016). The model parameters are tuned based on the development sets. We learn 200-dimensional word embeddings with Skip-gram model (Mikolov et al., 2013) on in-domain corpus, which follows (Tang et al., 2015a). The pre-trained word embeddings are used to initialize the embedding matrices $\mathbf{E}^A$ and $\mathbf{E}^B$. The dimensions of all hidden vectors are set to 200. For TripAdvisor dataset, the hop numbers of word-level and sentence-level iterative attention modules are set to 4 and 2 respectively. For BeerAdvocate dataset, the hop numbers are set to 6 and 2. The number of selected keywords $N_k = N$ is set to 20. To avoid model over-fitting, we use dropout and regularization as follows: (1) the regularization parameter is set to *1e-5*; (2) the dropout rate is set to 0.3, which is applied to both sentence and document vectors. All parameters are trained by ADADELTA (Zeiler, 2012) without needing to set the initial learning rate. To ensure fair comparisons, we make baselines have same settings as the proposed model, such as word embeddings, dimensions of hidden vectors and optimization details and so on.

3.4 Results and Analyses

We use accuracy and mean squared error (MSE) as the evaluation metrics and the results are shown in Table 2.

Model	TripAdvisor				BeerAdvocate			
	Dev		Test		Dev		Test	
	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE
Majority	24.47	2.533	23.89	2.549	24.48	4.706	24.41	4.545
SVM	34.30	1.982	35.26	1.963	25.70	3.286	25.79	3.270
SLDA	31.58	2.131	32.81	2.110	25.39	3.372	25.73	3.391
NBoW	38.43	1.866	39.09	1.808	28.99	2.883	28.85	2.919
DAN	40.30	1.569	40.93	1.531	31.25	2.569	32.44	2.279
CNN	43.25	1.474	43.35	1.456	34.17	2.173	33.37	2.217
LSTM	43.85	1.525	44.02	1.470	35.23	2.112	34.78	2.097
HAN	44.47	1.312	44.68	1.301	36.57	1.903	36.03	1.920
MHDAN	42.22	1.554	42.47	1.549	32.76	2.358	32.54	2.376
MHCNN	44.19	1.329	43.79	1.398	36.10	1.966	35.33	1.976
MHLSTM	44.53	1.308	44.72	1.272	38.14	1.785	37.04	1.809
MHAN	44.72	1.294	44.94	1.210	37.98	1.783	36.82	1.813
Our	46.21	1.091	**46.56**	**1.083**	39.43	1.696	**38.06**	**1.755**

Table 2: Comparison of our model and other baseline methods.

Model	TripAdvisor		BeerAdvocate	
	Accuracy	MSE	Accuracy	MSE
MHLSTM	44.75 (0.24)	1.256 (0.05)	37.28 (0.43)	1.802 (0.17)
MHAN	45.02 (0.33)	1.221 (0.12)	37.02 (0.22)	1.810 (0.15)
Our	**46.65**†(0.29)	**1.084***(0.06)	**38.25**†(0.35)	**1.749***(0.18)

Table 3: The results of average accuracy/MSE and standard deviation of models on test sets. We choose *MHAN* and *MHLSTM* as comparison baselines for TripAdvisor and BeerAdvocate respectively. In t-tests, the marker * refers to p-value < 0.05 and the marker † refers to p-value < 0.01.

Compared to SVM and SLDA, NBoW achieves higher accuracy by 3% in both datasets, which shows that embedding features are more effective than traditional ngram features on these two datasets. All neural network models outperform NBoW. It shows the advantages of neural networks in the document sentiment classification.

From the results of neural networks, we can observe that DAN performs worse than LSTM and CNN, and LSTM achieves slightly higher results than CNN. It can be explained that the simple composition method averaging embeddings of words in a document but ignoring word order, may not be as effective as other flexible composition models, such as LSTM and CNN, on aspect classification. Additionally, we observe that the multi-task learning and hierarchical architecture are beneficial for neural networks. Among all baselines, MHAN and MHLSTM achieve comparable results and outperform others.

Compared with MHAN and MHLSTM, our method achieves improvements of 1.5% (3% relative improvement) and 1.0% (2.5% relative improvement) on TripAdvisor and BeerAdvocate respectively, which shows that the incorporation of iterative attention mechanism helps the deep neural network based model build up more discriminative aspect-aware representation. Note that BeerAdvocate is relatively more difficult since the predicted ratings are from 1 to 10 while TripAdvisor is 1 to 5. Moreover, t-test is conducted by randomly splitting datasets into train/dev/test sets and random initialization. The results on test sets are described in Table 3 which show performance of our model is stable.

3.5 Case Study for Attention Results

In this section, we sample two sentences from TripAdvisor to show the visualization of attention results for case study. Both word-level and sentence-level attention visualizations are shown in Figure 4. We normalize the word weight by the sentence weight to make sure that only important words in a document are highlighted.

From the top figures in (a) and (b), we observe that our model assigns different attention weights for each aspect. For example, in the first sentence, the words *comfortable* and *bed* are assigned higher

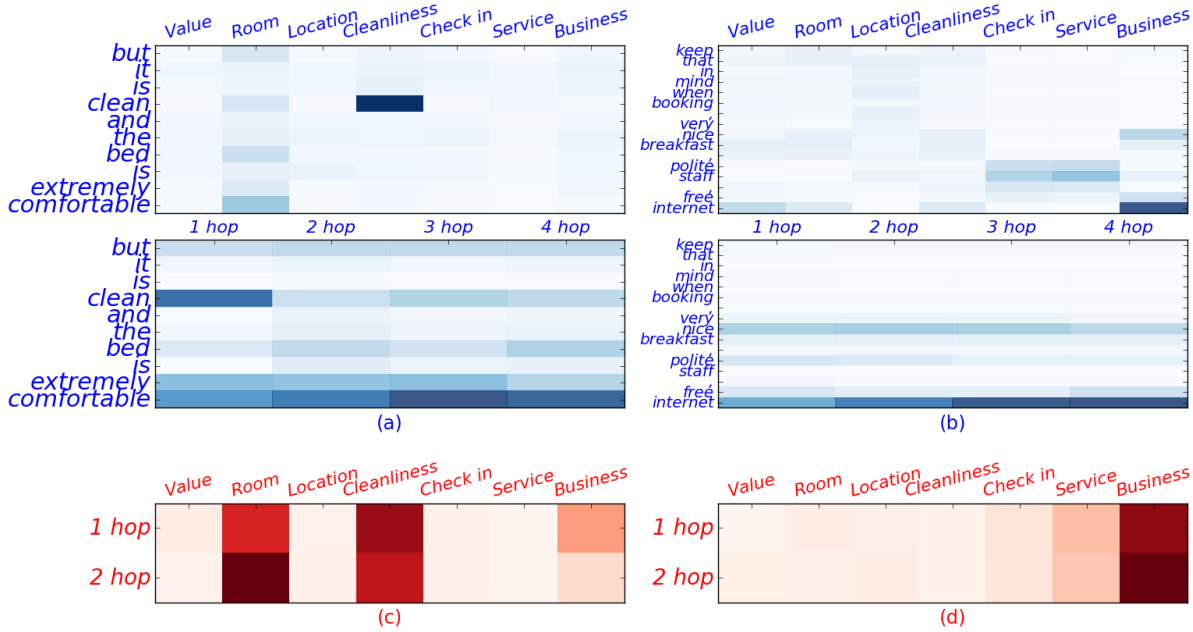

Figure 4: The attention visualization of words and sentences. Darker color means higher weight. (a) and (b) are the visualization of word weights; (c) and (d) are the visualization of sentence weights. The top figures in (a) and (b) show the word weights of fourth hop for each aspect. The bottom figures in (a) and (b) visualize the word weights of different hops for the aspects *Room* and *Business* respectively.

weights in the aspect *Room*, and the word *clean* are highlighted by the aspect *Cleaniness*. In the second sentence, the word *internet* is assigned a high attention value for *Business*. Moreover, the bottom figures in (a) and (b) show that (1) word weights of different hops are various; (2) attention values in higher hop are more reasonable. Specifically, in the first sentence, the weight of word *clean* is higher than the word *comfortable* in first hop, while *comfortable* surpasses *clean* in higher hops. In the second sentence, we observe that the value of word *internet* increases with the number of hop. Thus, we can see that the more sensible weights are obtained for words through the proposed iterative attention mechanism. Similarly, the figures (c) and (d) show that the conclusion from words is also suitable for sentences. For the first sentence, the sentence weight regarding the aspect *Room* is lower than *Cleanliness* in the first hop, but surpasses *Cleanliness* in the second hop. For the second sentence, the weight for *Business* becomes higher in the second hop.

3.6 Effects of Hop and Aspect Keywords

In this experiment, we investigate the effects of hop number m and size of aspect keywords N on performances. All the experiments are conducted on the development set. Due to lack of space, we only present the results of TripAdvisor and the results of BeerAdvocate have a similar behavior as TripAdvisor.

For the hop number, we vary m from 1 to 7 and the results are shown in Figure 5 (left). We can see that: (1) at the word level, the performance increases when $m \leq 4$, but shows no improvement after $m > 4$; (2) at the sentence level, model performs best when $m = 2$. Moreover, we can see that the hop number of word level leads to larger variation than the hop number of sentence level.

For the size of aspect keywords, we vary N from 0 to 35, incremented by 5. Note that, we set a learnable vector to represent question memory when $N = 0$. The results are shown in Figure 5 (right). We observe that the performance increases when $N \leq 20$, and has no improvement after $N > 20$. This indicates that a small number of keywords can help the proposed model achieve competitive results.

4 Related Work

Multi-Aspect Sentiment Classification. Multi-aspect sentiment classification has been studied extensively in literature. Lu et al. (2011) used support vector regression model based on hand-

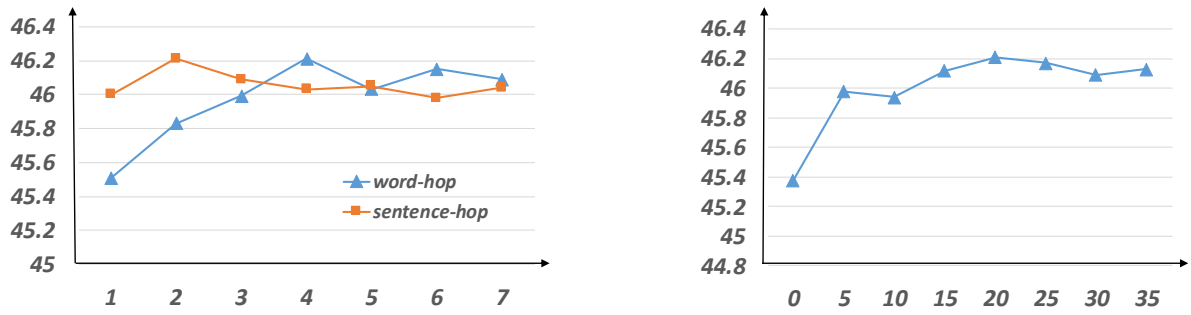

Figure 5: Results of different hops and different sizes of question keywords. Left: different the hop numbers; Right: different sizes of keywords.

crafted features to predict aspect ratings. To handle the correlation between aspects, McAuley et al. (2012) added a dependency term in final multi-class SVM objective. There were also some heuristic based methods and sophisticated topic models where multi-aspect sentiment classification is solved as a subproblem (Titov and McDonald, 2008; Wang et al., 2010; Diao et al., 2014; Pappas and Popescu-Belis, 2014). However, these approaches often rely on strict assumptions about words and sentences, for example, using the word syntax to determine if a word is an aspect or a sentiment word, or relating a sentence with an specific aspect. Another related problem is called aspect-based sentiment classification (Pontiki et al., 2014, 2016; Poria et al., 2016), which first extracts aspect expressions from sentences (Poria et al., 2014; Balahur and Montoyo, 2008; Chen et al., 2014, 2013), and then determines their sentiments. With the developments of neural networks and word embeddings in NLP, neural network based models have shown the state-of-the-art results with less feature engineering work. Tang et al. (2016) employed a deep memory network for aspect-based sentiment classification given the aspect location and Lakkaraju et al. (2014) employed recurrent neural networks and its variants for the task of extraction of aspect-sentiment pair. However, these tasks are sentence-level. Another related research field is document-level sentiment classification because we can treat single aspect sentiment classification as an individual document classification task. This line of research includes (Tang et al., 2015b; Chen et al., 2016; Tang et al., 2016; Yang et al., 2016) which are based on neural networks in a hierarchical structure. However, they did not work on multiple aspects.

Machine Comprehension. Recently, neural network based machine comprehension (or reading) has been studied extensively in NLP, with the releases of large-scale evaluation datasets (Hermann et al., 2015; Hill et al., 2016; Rajpurkar et al., 2016). Most of the related studies focus on attention mechanism (Bahdanau et al., 2014) which is firstly proposed in machine translating and aims to solve the long-distance dependency between words. Hermann et al. (2015) used Bi-LSTM to encode document and query, and proposed Attentive Reader and Impatient Reader. The first one attends document based on the query representation, and the second one attends document by the representation of each token in query with an incremental manner. Memory Networks (Weston et al., 2015; Sukhbaatar et al., 2015) attend and reason document representation in a multi-hop fashion, enriching interactions between documents and questions. Dynamic Memory Network (Kumar et al., 2016) updates memories of documents by re-running GRU models based on derived attention weights. Meanwhile, the query representation is refined by another GRU model. Gated-Attention Reader (Dhingra et al., 2016) proposes a novel attention mechanism, which is based on multiplicative interactions between the query embeddings and the intermediate states of a recurrent neural network document reader. Bi-Directional Attention Model (Xiong et al., 2017; Seo et al., 2017) fuses co-dependent representations of queries and documents in order to focus on relevant parts of both. Iterative Attention model (Sordoni et al., 2016) attends question and document sequentially, which is related to our model. Different from Iterative Attention model, our model focuses on the document-level multi-aspect sentiment classification, which is proposed

in a hierarchical architecture and has different procedures in the iterative attention module. Another related research problem is visual question answering which uses an image as question context rather than a set of keywords as question. Neural network based visual question answering (Lu et al., 2016; Xiong et al., 2016) is similar as the proposed models in text comprehension.

5 Conclusion

In this paper, we model the document-level multi-aspect sentiment classification as a text comprehension problem and propose a novel hierarchical iterative attention model in which documents and pseudo aspect-questions are interleaved at both word and sentence-level to learn aspect-aware document representation in a unified model. Extensive experiments show that our model outperforms the other neural models with multi-task framework and hierarchical architecture.

6 Acknowledgments

This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Nos. 61472006 and 91646202) as well as the National Basic Research Program (973 Program No. 2014CB340405). This work was also supported by NVIDIA Corporation with the donation of the Titan X GPU, Hong Kong CERG Project 26206717, China 973 Fundamental R&D Program (No.2014CB340304), and the LORELEI Contract HR0011-15-2-0025 with DARPA. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We also thank the anonymous reviewers for their valuable comments and suggestions that help improve the quality of this manuscript.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Alexandra Balahur and Andres Montoyo. 2008. A feature dependent method for opinion mining and classification. In *Natural Language Processing and Knowledge Engineering*. pages 1–7.

David M. Blei and Jon D. Mcauliffe. 2010. Supervised topic models. *Advances in Neural Information Processing Systems* 3:327–332.

Rich Caruana. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of EMNLP*. pages 1650–1659.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL*. pages 347–358.

Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Exploiting domain knowledge in aspect extraction. In *EMNLP*. pages 1655–1667.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Bhuwan Dhingra, Hanxiao Liu, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549* .

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of KDD*. ACM, pages 193–202.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research* 9(Aug):1871–1874.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NIPS*. pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *Proceedings of ICLR*.

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

Mohit Iyyer, Varun Manjunatha, Jordan L Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of ACL*. pages 1681–1691.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*. pages 1746–1751.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of ICML*.

Himabindu Lakkaraju, Richard Socher, and Chris Manning. 2014. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP*. Austin, Texas, pages 107–117.

Bing Liu. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition.*, pages 627–666.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *ICDM Workshops*. IEEE, pages 81–88.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Proceedings of NIPS*. pages 289–297.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of ICLR*.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of ICDM*. IEEE, pages 1020–1025.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*. pages 3111–3119.

Bo Pang and Lillian Lee. 2007. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2):1–135.

Nikolaos Pappas and Andrei Popescu-Belis. 2014. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of EMNLP*. pages 455–466.

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. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval*. pages 19–30.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*. pages 27–35.

Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui, and Alexander Gelbukh. 2014. A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*. pages 28–37.

Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Federica Bisio. 2016. Sentic lda: Improving on lda with semantic similarity for aspect-based sentiment analysis. In *International Joint Conference on Neural Networks*. pages 4465–4473.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*. pages 2383–2392.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *Proceedings of ICLR* .

Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245* .

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Proceedings of NIPS*. pages 2440–2448.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*. pages 1422–1432.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *ACL*. pages 1014–1023.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP*. pages 214–224.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688. http://arxiv.org/abs/1605.02688.

Ivan Titov and Ryan T McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL*. Citeseer, volume 8, pages 308–316.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of KDD*. ACM, pages 783–792.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *Proceedings of ICLR*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of ICML*. pages 1378–1387.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. *Proceedings of ICLR* .

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*. pages 1480–1489.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .