Human-Like Decision Making: Document-level Aspect Sentiment Classification via Hierarchical Reinforcement Learning

Jingjing Wang[1], **Changlong Sun**[2], **Shoushan Li**[1,*], **Jiancheng Wang**[1], **Luo Si**[2], **Min Zhang**[1], **Xiaozhong Liu**[2], **Guodong Zhou**[1]
[1]School of Computer Science and Technology, Soochow University, China
[2]Alibaba Group, China
{djingwang, lishoushan, minzhang, gdzhou}@suda.edu.cn,
jiancheng.wang@qq.com, {changlong.scl, xiaozhong.lxz, luo.si}@alibaba-inc.com

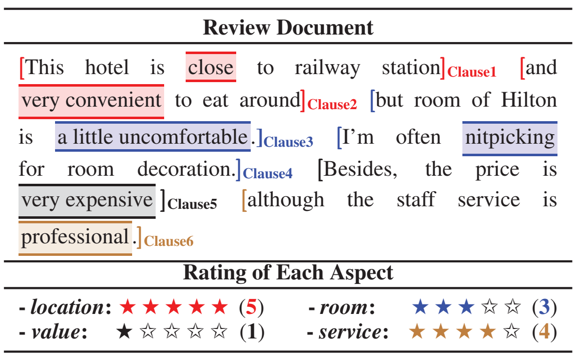
Abstract

Recently, neural networks have shown promising results on Document-level Aspect Sentiment Classification (DASC). However, these approaches often offer little transparency w.r.t. their inner working mechanisms and lack interpretability. In this paper, to simulating the steps of analyzing aspect sentiment in a document by human beings, we propose a new Hierarchical Reinforcement Learning (HRL) approach to DASC. This approach incorporates clause selection and word selection strategies to tackle the data noise problem in the task of DASC. First, a high-level policy is proposed to select aspect-relevant clauses and discard noisy clauses. Then, a low-level policy is proposed to select sentiment-relevant words and discard noisy words inside the selected clauses. Finally, a sentiment rating predictor is designed to provide reward signals to guide both clause and word selection. Experimental results demonstrate the impressive effectiveness of the proposed approach to DASC over the state-of-the-art baselines.

1 Introduction

Document-level Aspect Sentiment Classification (DASC) is a fine-grained sentiment classification task in the field of sentiment analysis (Pang and Lee, 2007; Li et al., 2010). This task aims to predict the sentiment rating for each given aspect mentioned in a document-level review. For instance, Figure 1 shows a review document with four given aspects of a hotel (i.e., *location*, *room*, *value*, *service*). The goal of DASC is to predict the rating score towards each aspect by analyzing the whole document. In the last decade, this task has been drawing more and more interests of researchers in the Natural Language Processing community (Titov and McDonald, 2008; Yin et al., 2017; Li et al., 2018). In previous studies, neu-

*Corresponding author



Review Document

[This hotel is close to railway station]Clause1 [and very convenient to eat around]Clause2 [but room of Hilton is a little uncomfortable.]Clause3 [I'm often nitpicking for room decoration.]Clause4 [Besides, the price is very expensive]Clause5 [although the staff service is professional.]Clause6

Rating of Each Aspect

- *location*:	★★★★★ (5)	- *room*:	★★★☆☆ (3)
- *value*:	★☆☆☆☆ (1)	- *service*:	★★★★☆ (4)

Figure 1: An example of a review document, where clauses and words with different colors refer to different aspects.

ral models have shown to be effective for performance improvement on DASC. Despite the advantages, these complex neural network approaches often offer little transparency w.r.t. their inner working mechanisms and suffer from the lack of interpretability. However, clearly understanding where and how such a model makes such a decision is rather important for developing real-world applications (Liu et al., 2018; Marcus, 2018).

As human beings, if asked to evaluate the sentiment rating for a specific aspect in a document, we often perform sentiment prediction in two steps. First, we select some **aspect-relevant snippets** (e.g., sentences/clauses) inside the document. Second, we select some **sentiment-relevant words** (e.g., sentiment words) inside these snippets to make a rating decision. For instance, for aspect *location* in Figure 1, we first select the aspect-relevant clauses, i.e., Clause1 and Clause2, and then select sentiment-relevant words, i.e., "*close*" and "*very convenient*" inside the two clauses, for making the rating decision (5 stars).

Inspired by the above cognitive process of human beings, one ideal and interpretable solution for DASC is to select aspect-relevant clauses and sentiment-relevant words, discarding those noisy parts of a document for decision making. In this solution, two major challenges exist which are illustrated as follows.

The first challenge is how to select aspect-relevant clauses and discard those irrelevant and noisy clauses. For instance, for aspect *location*, Clause5 mentioning another aspect *value* (only 1 star) may induce the noise and should be discarded, because the noise can provide wrong signals to mislead the model into assigning very low sentiment rating to aspect *location*. One possible way to alleviate this noisy problem is to leverage the soft-attention mechanism as proposed in Li et al. (2018) and Wang et al. (2018). However, this soft-attention mechanism has the limitation that the *softmax* function always assigns small but non-zero probabilities to noisy clauses, which will weaken the attention given to the few truly significant clauses for a particular aspect. Therefore, a well-behaved approach should discard noisy clauses for a specific aspect during model training.

The second challenge is how to select sentiment-relevant words and discard those irrelevant and noisy words. For instance, for aspect *location*, words "*this*", "*is*" in Clause1 are noisy words and should be discarded since they make no contribution to implying the sentiment rating. One possible way to alleviate this problem is to also leverage the soft-attention mechanism as proposed in Li et al. (2018). However, this soft-attention mechanism may induce additional noise and lack interpretability because it tends to assign higher weights to some domain-specific words rather than real sentiment-relevant words (Mudinas et al., 2012; Zou et al., 2018). For instance, this soft-attention mechanism tends to regard the name of a hotel "*Hilton*" with a good reputation in Clause3 as a positive word which could mislead the model into assigning a higher rating to aspect *room*. Therefore, a well-behaved approach should highlight sentiment-relevant words and discard noisy words for a specific aspect during model training.

In this paper, we propose a Hierarchical Reinforcement Learning (HRL) approach with a high-level policy and a low-level policy to address the above two challenges in DASC. First, a high-level policy is leveraged to select aspect-relevant clauses and discard noisy clauses during model training. Then, a low-level policy is leveraged to select sentiment-relevant words and discard noisy words inside the above selected clauses. Finally, a sentiment rating predictor is designed to provide reward signals to guide both clause and word selection. The empirical studies show that the proposed approach performs well by incorporating the clause selection and word selection strategies and significantly outperforms several state-of-the-art approaches including those with the soft-attention mechanism.

2 Hierarchical Reinforcement Learning

Figure 2 shows the overall framework of our Hierarchical Reinforcement Learning (HRL) approach which contains three components: a high-level policy for clause selection (Section 2.2); a low-level policy for word selection (Section 2.3); a sentiment rating predictor for providing reward signals to guide both the above clause and word selection (Section 2.4).

As a preprocessing, we adopt RST style discourse segmentation[1] (MANN, 1988) to segment all documents in corpus $\mathcal{C}$ into Elementary Discourse Units (EDUs), and consider thse EDUs as clauses by following Wang et al. (2018).

In summary, we formulate the task of DASC as a semi-Markov Decision process (Sutton et al., 1999b), i.e., hierarchical reinforcement learning with a high-level policy and a low-level policy. In particular, our HRL approach for DASC works as follows. Given a review document with a clause sequence and an aspect, the high-level policy decides whether a clause mentions this aspect. If yes, the high-level policy selects this clause and launches the low-level policy, which scans the words inside this selected clause one by one in order to select sentiment-relevant words. Otherwise, the high-level policy skips current clause and turns to the next clause until all clauses in the review document are scanned. During clause and word selection, a sentiment rating predictor is employed to provide reward signals to guide the above clause and word selection.

2.1 Clause Selection with High-level Policy

Assume that a review document $\mathcal{D}$ with a given aspect x_{aspect} has been segmented into a clause sequence $\{u_1, ..., u_n\}$, high-level policy π^h aims

[1]In preliminary experiments, we tried directly adopting sentence segmentation by following Li et al. (2018) rather than clause splitting and found that adopting clause splitting achieves better performance. Detailed comparison results are presented in ablation study of Section 3.2.

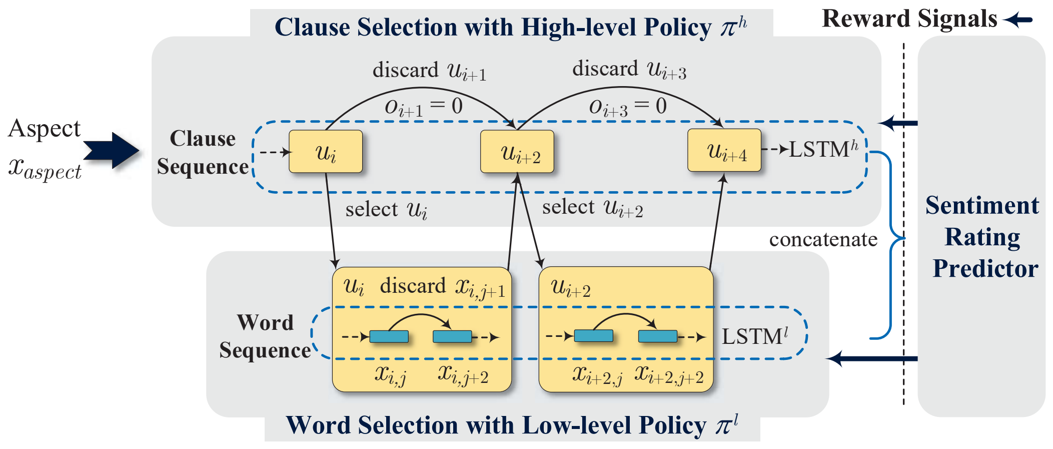


Figure 2: Overall architecture of our Hierarchical Reinforcement Learning (HRL) approach.

to select clause u_i which truly mentions aspect x_{aspect} and discard noisy ones. Here, clause u_i consists of k_i words $\{x_{i,1}, ..., x_{i,k_i}\}$. Once a clause is selected, it is passed to the low-level policy for further word selection.

During clause selection, we adopt a stochastic policy as high-level policy π^h, which can generate a conditional probability distribution $\pi^h(o|\cdot)$ over option (i.e., high-level action) sequence $o = [o_1, ..., o_n]$. Here, $o_i = 1$ indicates u_i is selected; otherwise $o_i = 0$ indicates u_i is discarded. More specifically, we adopt a LSTM model LSTM^h to construct high-level policy π^h for performing clause selection over the clause sequence. In LSTM^h, the hidden state $\hat{v}_i \in \mathbb{R}^d$ of clause u_i and memory cell c_i^h at i-th time-step are given by,

$$\hat{v}_i, c_i^h = \begin{cases} f(\hat{v}_{i-1}, c_{i-1}^h, v_i) & o_i = 1 \\ \hat{v}_{i-1}, c_{i-1}^h & o_i = 0 \end{cases} \tag{1}$$

where v_i is the vector representation of clause u_i and initialized by hidden state $\hat{w}_{i,k_i}$ of the last word $\hat{x}_{i,k_i}$ in clause u_i. Here, $\hat{w}_{i,k_i}$ is obtained from the pre-trained LSTM^l (presented in line 3 of Algorithm 1); f denotes all gate functions and update function of LSTM^h. Note that if $o_i = 0$, LSTM^h will skip (i.e., discard) and not encode clause u_i, memory cell c_i^h and hidden state $\hat{v}_i$ of current time-step i are then directly copied from the previous time-step $i-1$.

In principle, the high-level policy π^h uses a **Reward** to guide clause selection over the clause sequence. It samples an **Option** o_i with the probability $\pi^h(o_i|s_i^h; \theta^h)$ at each **State** s_i^h. More concretely, the state, option and reward of π^h are defined as follows.

• **State.** The state s_i^h at i-th time-step should provide adequate information for deciding to select a clause or not for x_{aspect}. Thus, state $s_i^h \in \mathbb{R}^{4d}$ is composed of four parts, i.e., $\hat{v}_{i-1}$, c_{i-1}^h, v_i and v_a, defined as $s_i^h = \hat{v}_{i-1} \oplus c_{i-1}^h \oplus v_i \oplus v_a$, where $v_a \in \mathbb{R}^d$ is aspect embedding[2] of x_{aspect}.

• **Option.** π^h samples *option* $o_i \in \{0, 1\}$ by the conditional probability $\pi^h(o_i|s_i^h; \theta^h)$ which could be cast as a binary classification problem. Thus, we adopt logistic function to define $\pi^h(o_i|s_i^h; \theta^h)$.

$$\begin{aligned} o_i \sim \pi^h(o_i|s_i^h; \theta^h) &= o_i \sigma(W^h s_i^h + b^h) \\ &+ (1 - o_i)(1 - \sigma(W^h s_i^h + b^h)) \end{aligned} \tag{2}$$

where $\theta^h = \{W^h \in \mathbb{R}^{1\times 4d}, b^h \in \mathbb{R}\}$ is the trainable parameter; $\sim$ denotes sampling operation; σ denotes sigmod function.

• **Reward.** In order to select aspect-relevant clauses inside a clause sequence $\{u_1, ..., u_n\}$, given a sampled *option* trajectory $\tau^h = (s_1^h, o_1, r_1^h, ..., s_n^h, o_n, r_n^h) \sim \pi^h$, we compute the high-level cumulative reward r_i^h at i-th time-step as follows:

$$\begin{aligned} r_i^h &= \lambda_1 \sum_{t=i}^{n} \gamma^{t-i} \log \cos(v_a, \hat{v}_t) \\ &+ \lambda_2 \sum_{t=i}^{n} \gamma^{t-i} r^l(u_t) + \lambda_3 \log p_\theta(y|\hat{v}_n) \end{aligned} \tag{3}$$

where r_i^h consists of three different terms: **1)** The first term $\log \cos(v_a, \hat{v}_t)$ is a cosine intermediate reward computed by *cosine* similarity between aspect embedding $v_a \in \mathbb{R}^d$ and hidden state $\hat{v}_t \in \mathbb{R}^d$ of the t-th clause u_t. This reward provides aspect supervision signals to guide the policy to select aspect-relevant clauses. **2)** The second term

[2]Aspect embedding is initialized by an average of its keywords embeddings. Keywords for aspects (e.g., keywords for aspect *service* are service, smile etc.) are obtained from Yin et al. (2017).

$r^l(u_t) = \sum_{j=1}^{k_t} r_{t,j}^l$ is an intermediate reward from low-level policy after the word selection in the selected clause u_t is finished. Note that if clause u_t is discarded, $r^l(u_t) = 0$. This reward provides a feedback to indicate how well clause selection is. **3)** The third term $\log p_\theta(y|\hat{v}_n)$ is a delay reward from sentiment rating predictor. After LSTM^h finishes all options, we feed the last hidden state $\hat{v}_n$ of LSTM^h to the softmax decoder of sentiment rating predictor and then obtain a rating probability $p_\theta(y|\hat{v}_n)$ for ground-truth rating label y to compute this delay reward. This reward provides additional reward signals to guide policy to select discriminative clauses. Besides, γ is the discount factor; λ_1, λ_2 and λ_3 are weight parameters.

2.2 Word Selection with Low-level Policy

Given a word sequence $\{x_{i,1}, ..., x_{i,k_i}\}$ of clause u_i selected by the high-level policy, low-level policy π^l aims to select the sentiment-relevant word $x_{i,j}$ and discard noisy ones.

During word selection, we still adopt a stochastic policy as low-level policy π^l, which can generate a conditional probability distribution $\pi^l(a|\cdot)$ over action sequence $a = [a_{i,1}, ..., a_{i,k_i}]$, where $a_{i,j} = 1$ indicates j-th word x_{ij} in i-th clause is selected; $a_{i,j} = 0$ indicates $x_{i,j}$ is discarded. Similar to clause selection, we adopt another LSTM model LSTM^l to construct low-level policy π^l for performing word selection over word sequence $\{x_{i,1}, ..., x_{i,k_i}\}$ of each clause (Note that, as shown in Figure 2, LSTM^l is shared by all selected clauses from high-level policy). In LSTM^l, the hidden state $\hat{w}_{i,j} \in \mathbb{R}^d$ of word $x_{i,j}$ and memory cell c_j^l at j-th time-step (Here, we omit the clause index and only use j to denote the j-th time-step in i-th clause u_i) are given by,

$$\hat{w}_{i,j}, c_{i,j}^l = \begin{cases} f(\hat{w}_{i,j-1}, c_{i,j-1}^l, w_{i,j}) & a_{i,j} = 1 \\ \hat{w}_{i,j-1}, c_{i,j-1}^l & a_{i,j} = 0 \end{cases} \tag{4}$$

where $w_{i,j}$ is word embedding of $x_{i,j}$. Similar to LSTM^h, in LSTM^l, if $a_{i,j} = 0$, i.e., word $x_{i,j}$ is discarded, memory cell and hidden state of current time-step are directly copied from the previous time-step. Then, we illustrate state, action and reward of low-level policy as follows.

• **State.** The state $s_{i,j}^l$ at j-th time-step should provide adequate information for deciding to select a word or not. Thus, the state $s_{i,j}^l \in \mathbb{R}^{3d}$ is composed of three parts, i.e., $\hat{w}_{i,i-1}$, $c_{i,j-1}^l$, $w_{i,j}$, defined as $s_{i,j}^l = \hat{w}_{i,j-1} \oplus c_{i,j-1}^l \oplus w_{i,j}$.

• **Action.** π^l samples *action* $a_{i,j} \in \{0, 1\}$ by the conditional probability $\pi^l(a_{i,j}|s_{i,j}^l; \theta^l)$. Thus, similar to high-level policy, we adopt logistic function to define $\pi^l(a_{i,j}|s_{i,j}^l; \theta^l)$.

$$\begin{aligned} a_{i,j} \sim \pi^l(a_{i,j}|s_{i,j}^l; \theta^l) = a_{i,j}\sigma(W^l s_{i,j}^l + b^l) \\ +(1 - a_{i,j})(1 - \sigma(W^l s_{i,j}^l + b^l)) \end{aligned} \tag{5}$$

where $\theta^l = \{W^l \in \mathbb{R}^{1\times 3d}, b^l \in \mathbb{R}\}$ is the trainable parameter.

• **Reward.** Similarly, in order to select sentiment-relevant words inside a word sequence $\{x_{i,1}, ..., x_{i,k_i}\}$, given a sampled *action* trajectory $\tau^l = (s_{i,1}^l, a_{i,1}, r_{i,1}^l, ..., s_{i,k_i}^l, a_{i,k_i}, r_{i,k_i}^l) \sim \pi^l$, we compute the low-level cumulative reward $r_{i,j}^l$ at j-th time-step as:

$$r_{i,j}^l = \lambda_1' \log p_\theta(y|\hat{w}_{i,k_i}) + \lambda_2'(-N')/N \tag{6}$$

where $r_{i,j}^l$ consists of two terms: **1)** Similar to high-level policy, the first term $\log p_\theta(y|\hat{w}_{i,k_i})$ is a delay reward provided by sentiment rating predictor. After LSTM^l finishes all actions, we feed last hidden state $\hat{w}_{i,k_i}$ in i-th clause to softmax decoder of sentiment rating predictor and then we can obtain this delay reward. This reward provides rating supervision information to guide policy to select discriminative words, i.e., sentiment-relevant words. **2)** The second term $\gamma(-N')/N$ is a penalty delay reward. $N' = \sum_{j=1}^{k_i} a_{i,j}$ denotes the number of selected words. The basic idea of using this penalty reward is to select words as small as possible because sentiment-relevant words is usually a small subset of all words inside the clause. Note that, we could also adopt external sentiment lexicons to achieve this goal, but sentiment lexicons are difficult to obtain in many real-world applications. Besides, λ_1', λ_2' are weight parameters.

2.3 Sentiment Rating Predictor

The goal of sentiment rating predictor lies in two-fold. On one hand, during model training, the goal of sentiment rating predictor is to use a softmax decoder to provide rating probabilities as the reward signals (see Eq.(3) and Eq.(6)) to guide both clause and word selection.

On the other hand, when model training is finished, i.e., both high-level and low-level policy finish all their selections, the goal of sentiment rating predictor is to perform DASC. Specifically, we first regard last state $\hat{v}_n$ of LSTM^h as the representation of all selected clauses while last state

$\hat{w}_{n,k_n}$ of LSTM^l as the representation of all selected words. Then, we concatenate $\hat{v}_n$ and $\hat{w}_{n,k_n}$ to compute the final representation z of the review document $\mathcal{D}$ as: $z = \hat{v}_n \oplus \hat{w}_{n,k_n}$, where $\oplus$ denotes concatenation operation. Finally, to perform DASC, we feed z to a softmax decoder as follows:

Softmax Decoder. We first feed z to a softmax classifier $m = Wz + b$, where $m \in \mathbb{R}^C$ is output vector; $\theta = \{W, b\}$ is trainable parameter. Then, the probability of labeling sentence with sentiment rating $\hat{y} \in [1, C]$ is computed by $p_\theta(\hat{y}|z) = \frac{\exp(m_{\hat{y}})}{\sum_{\ell=1}^{C}\exp(m_\ell)}$. Finally, the label with the highest probability stands for the predicted sentiment rating for x_{aspect}.

Note that, for an aspect, if no clauses and words are finally selected from a review document, the model will assign a random rating for this aspect.

2.4 Model Training via Policy Gradient and Back-Propagation

The parameters in HRL are learned according to Algorithm 1. Specifically, these parameters can be divided into two groups: **1)** θ^h and θ^l of high-level policy π^h and low-level policy π^l respectively. **2)** θ of LSTM^h, LSTM^l and softmax decoder.

For θ^h of high-level policy, we optimize it with policy gradient (REINFORCE) (Williams, 1992; Sutton et al., 1999a). The policy gradient w.r.t. θ^h is computed by differentiating the maximized expected reward $J(\theta^h)$ as follows:

$$\nabla_{\theta^h} J(\theta^h) = \mathbb{E}_{\tau^h \sim \pi^h}[\sum_{i=1}^{n} \mathcal{R}^h \nabla_{\theta^h} \log \pi^h(o_i|s_i^h; \theta^h)] \quad (7)$$

where $\mathcal{R}^h = r_i^h - b(\tau^h)$ is the advantage estimate of the high-level reward. Here, $b(\tau^h)$ is the *baseline* (Williams, 1992) which is used to reduce the variance of the high-level reward without altering its expectation theoretically. In practical use, we sample some trajectories $\tau_1^h, \tau_2^h, ..., \tau_m^h$ over the clause sequence with the current high-level policy. The model will assign a reward score to each sequence according to the designed scores function, and then estimates $b(\tau^h)$ as the average of those rewards. Similarly, the policy gradient w.r.t. θ^l of low-level policy is given by,

$$\nabla_{\theta^l} J(\theta^l) = \mathbb{E}_{\tau^l \sim \pi^l}[\sum_{j=1}^{k_i} \mathcal{R}^l \nabla_{\theta^l} \log \pi^l(a_{i,j}|s_{i,j}^l; \theta^l)] \quad (8)$$

Algorithm 1 Hierarchical reinforcement learning

```
1: Input: Corpus $\mathcal{C}$; a review document $\mathcal{D}$ with a clause sequence $\{u_1, ..., u_n\}$; a clause $u_i$ with a word sequence $\{x_{i,1}, ..., x_{i,k_i}\}$.
2: Initialize parameters $\theta$, $\theta^h$ and $\theta^l$ randomly;
3: Pre-train LSTM$^l$ by forcing $\pi^l$ to select all words for classification, and update $\theta$ of LSTM$^l$ by Eq.(9);
4: Pre-train LSTM$^h$ by forcing $\pi^h$ to select all clauses for classification, and update $\theta$ of LSTM$^h$ by Eq.(9);
5: Fix all parameters $\theta$ and update $\theta^h$, $\theta^l$ as follows:
6: for review document $\mathcal{D} \in \mathcal{C}$ do
7:   for clause $u_i \in \{u_1, ..., u_n\}$ do
8:     Sample option $o_i \sim \pi^h(o_i|s_i^h; \theta^h)$;
9:     if option $o_i = 1$ then
10:      for word $x_{i,j} \in \{x_{i,1}, ..., x_{i,k_i}\}$ do
11:        Sample action $a_{i,j} \sim \pi^h(a_{i,j}|s_{i,j}^l; \theta^h)$;
12:      end for
13:      Compute $r_{i,j}^l$ by Eq.(6);
14:      Update $\theta^l$ by Eq.(8);
15:    end if
16:  end for
17:  Compute $r_i^h$ by Eq.(3);
18:  Update $\theta^h$ by Eq.(7);
19: end for
```

Datasets	*#documents*	*#words/doc*	*#Aspect*
TripUser	58632	181.03	7
TripAdvisor	29391	251.7	7
BeerAdvocate	51020	144.5	4

Table 1: Statistics of three datasets. *#words/doc* denotes the number of words (average per document). *#Aspect* denotes the number of aspects in each dataset. The rating scales of TripAdvisor and BeerAdvocate are 1-5 and 1-10 respectively.

where $\mathcal{R}^l = r_{i,j}^l - b(\tau^l)$ is the advantage estimate of the low-level reward. Similarly, $b(\tau^l)$ is used to reduce the variance of the low-level reward.

For θ, we optimize it with back-propagation. The objective of learning θ is to minimize the cross-entropy loss in the classification phase, i.e.,

$$J(\theta) = \mathbb{E}_{(\mathcal{D}, x_{aspect}, y) \sim \mathcal{C}}[-\log p_\theta(y|z)] + \frac{\delta}{2}||\theta||_2^2 \quad (9)$$

where $(\mathcal{D}, x_{aspect}, y)$ denotes a review document $\mathcal{D}$ with a given aspect x_{aspect} from corpus $\mathcal{C}$; y is ground-truth sentiment rating for aspect x_{aspect}. δ is a L_2 regularization.

3 Experimentation

3.1 Experimental Settings

Data. We conduct our experiments on three public datasets on DASC, i.e., TripUser (Li et al., 2018), TripAdvisor (Wang et al., 2010) and BeerAdvocate (McAuley et al., 2012; Lei et al., 2016). In the experiment, we adopt Discourse Segmentation

Approaches	TripUser				TripAdvisor				BeerAdvocate			
	Development		*Test*		*Development*		*Test*		*Development*		*Test*	
	Acc.↑	MSE↓	Acc.↑	MSE↓	Acc.↑	MSE↓	Acc.↑	MSE↓	Acc.↑	MSE↓	Acc.↑	MSE↓
SVM	-	-	46.35†	1.025†	34.30‡	1.982‡	35.26‡	1.963‡	25.70‡	3.286‡	25.79‡	3.270‡
LSTM	53.23	0.787	52.74	0.794	43.85‡	1.525‡	44.02‡	1.470‡	35.23‡	2.112‡	34.78‡	2.097‡
MAMC	-	-	55.49†	0.583†	46.21‡	1.091‡	46.56‡	1.083‡	39.43‡	1.696‡	38.06‡	1.755‡
HARN	-	-	58.15†	0.528†	-	-	48.21‡	0.923‡	39.81	1.672	38.19	1.751
HUARN	-	-	60.70†	0.514†	-	-	-	-	-	-	-	-
C-HAN	58.49	0.602	57.38	0.543	47.61	0.914	47.08	0.955	38.67	1.703	37.95	1.801
HS-LSTM	59.75	0.566	59.01	0.524	48.45	0.947	46.84	1.013	37.43	1.870	36.83	1.912
RL-Word-Selection	60.15	0.475	59.55	0.519	48.55	0.913	48.51	0.917	39.92	1.648	38.45	1.697
RL-Clause-Selection	61.32	0.433	60.54	0.461	51.05	0.762	50.02	0.781	41.39	1.505	39.76	1.622
HRL	**62.97**	**0.336**	**62.84**	**0.351**	**52.71**	**0.652**	**52.27**	**0.662**	**43.41**	**1.416**	**41.39**	**1.503**

Table 2: Comparison of our approaches and other baseline approaches to DASC. The results with symbol † are retrieved from Li et al. (2018) and those with ‡ are from Yin et al. (2017)

Tool[3] to segment all reviews in the three datasets into EDUs (i.e., clauses). Moreover, we adopt training/development/testing settings (8:1:1) by following Yin et al. (2017); Li et al. (2018). Table 1 shows the statistics of the three datasets.

Implementation Details. We adopt the pre-trained 200-dimension word embeddings provided by Yin et al. (2017). The dimension of LSTM hidden states is set to be 200. The other hyper-parameters are tuned according to the performance in the development set. Specifically, we adopt Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.012 for cross-entropy training and adopt SGD optimizer with a learning rate of 0.008 for all policy gradients training. For rewards of high-level and low-level policies, γ is 0.8; λ_1, λ_2 and λ_3 are 0.25, 0.25 and 0.5 respectively. λ'_1, λ'_2 are 0.6 and 0.4. Additionally, the batch size is set to be 64, regularization weight is set to be 10^{-5} and the dropout rate is 0.2.

Evaluation Metrics. The performance is evaluated using Accuracy (Acc.) and MSE as Yin et al. (2017). Moreover, t-test is used to evaluate the significance of the performance difference between two approaches (Yang and Liu, 1999).

Baselines. We compare HRL with the following baselines: **1) SVM** (Yin et al., 2017). This approach only adopts unigram, bigram as features to train an SVM classifier. **2) LSTM** (Tang et al., 2015). This is a neural network approach to document-level sentiment classification which employs gated LSTM to learn text representation. **3) MAMC** (Yin et al., 2017). This approach employs hierarchical iterative attention to learn aspect-specific representation. This is a state-of-the-art approach to DASC. **4) HARN** (Li et al., 2018). This approach adopts hierarchical attention to incorporate overall rating and aspect information so as to learn aspect-specific representation. This is another state-of-the-art approach to DASC. **5) HUARN** (Li et al., 2018). This approach extends HARN by integrating additional user information. This is another state-of-the-art approach to DASC. **6) C-HAN** (Wang et al., 2018). This approach adopts hierarchical attention to incorporate clause and aspect information so as to learn text representation. Although this is a state-of-the-art approach to sentence-level ASC, it could also be directly applied in DASC. **7) HS-LSTM** (Zhang et al., 2018). This is a reinforcement learning approach to text classification, which employs a hierarchically LSTM to learn text representation. **8) RL-Word-Selection.** Our approach which leverages only the word selection strategy by using the low-level policy. **9) RL-Clause-Selection.** Our approach which leverages only the clause selection strategy by using the high-level policy.

3.2 Experimental Results

Table 2 shows the performance comparison of different approaches. From this table, we can see that, all LSTM-based approaches outperform **SVM**, showing that LSTM has potentials in automatically learning text representations and can bring performance improvement for DASC.

Four state-of-the-art ASC approaches including **MAMC**, **HARN**, **HUARN** and **C-HAN** all perform better than **LSTM**. These results confirm the helpfulness of considering aspect information in DASC. Besides, we find the reinforcement learning based approach **HS-LSTM** without consid-

[3]http://alt.qcri.org/tools/discourse-parser/

Approaches	TripUser		TripAdvisor		BeerAdvocate	
	Acc.↑	MSE↓	Acc.↑	MSE↓	Acc.↑	MSE↓
HRL	**62.84**	**0.351**	**52.27**	**0.662**	**41.39**	**1.503**
w/o *cosine* intermediate reward in Eq.(3)	60.15	0.487	49.02	0.811	38.72	1.673
w/o penalty delay reward in Eq.(6)	61.52	0.425	51.15	0.711	40.13	1.577
w/o the representation of selected clauses	58.40	0.622	46.33	1.207	38.61	1.684
w/o the representation of selected words	60.08	0.497	48.94	0.886	39.92	1.626
using sentence splitting instead of clause	59.74	0.504	50.11	0.842	39.33	1.651

Table 3: Ablation study of HRL on three different datasets.

ering aspect information can achieve comparable performance with **MAMC**, **HARN**, **C-HAN**, and even beat **MAMC** on two datasets **TripUser** and **TripAdvisor**, which demonstrates that using reinforcement learning is a good choice to learn text representation for DASC.

Our approach **RL-Word-Selection** and **RL-Clause-Selection** outperform most above approaches and they only perform slightly worse than **HUARN**. This result encourages to perform clause or word selection in DASC. Among all these approaches, our approach **HRL** performs best and it significantly outperforms (p-value $<$ 0.01) strong baseline **HUARN** which actually considers some other kinds of external information, such as the overall rating and the user information. These results encourage to perform both clause and word selection in DASC.

Ablation Study. Further, we conduct the ablation study of HRL to evaluate the contribution of each component. The results are shown in Table 3. From this table, we can see that **1)** Using *cosine* intermediate reward in Eq.(3) can averagely improve Acc. by 2.87% on three different datasets. **2)** Using penalty delay reward in Eq.(6) can averagely improve Acc. by 1.23%. **3)** To concatenate additional representation of the selected clauses in rating predictor can improve Acc. by 4.38%. **4)** To concatenate additional representation of the selected words in rating predictor can improve Acc. by 2.72%. **5)** Using the clause splitting instead of sentence splitting could improve Acc. by 2.44%. This confirms that it is more appropriate to consider clauses as the segmentation units than sentences. This is because that 90% of clauses contain only one opinion expression as proposed in Bayoudhi et al. (2015). For instance, as shown in Figure 1, if we use the sentences as the segmentation units, Clause1-Clause3 will be assigned into one unit while they talk about two aspects, i.e., *location* and *room*. In this scenario, sentence selection will not be able to discard noisy parts inside the sentence for the aspect *location* or *room*.

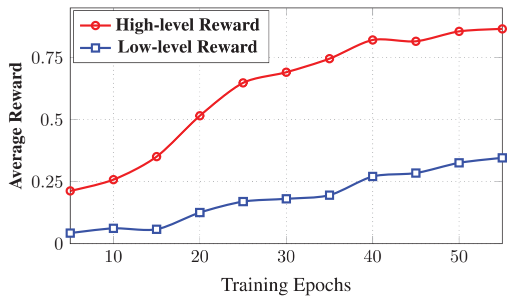


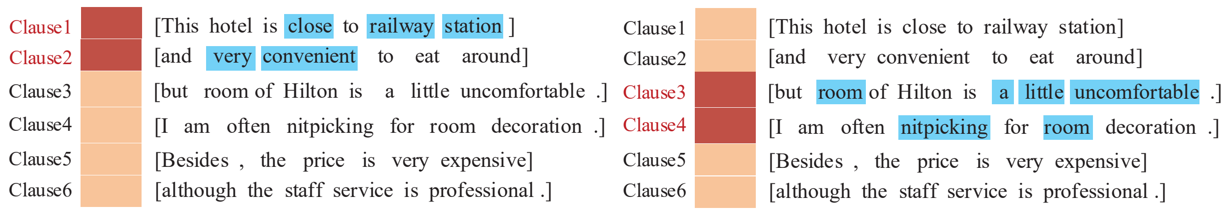
Figure 3: Training reward with different epochs

4 Analysis and Discussion

Analysis of HRL Training. Figure 3 shows two average rewards (each epoch) of high-level and low-level policy on BeerAdvocate respectively. To clearly observe the change of the reward, following Lillicrap et al. (2016), all rewards are normalized to $(0, 1)$. From this figure, we can see that, both the high-level and low-level reward increase as the training algorithm iterates. This result indicates that our HRL approach is capable of stably revising its policies to obtain more discriminative clauses and words for better performing.

Analysis of Clause and Word Selection. Figure 4 shows visualizations of our HRL approach which performs the clause selection and word selection on a review document. From this figure, we can see that HRL is able to precisely select aspect-relevant clauses, i.e., Clause1 and Clause2, for aspect *location* while select Clause3 and Clause4 for *room*. Further, HRL is able to select all sentiment-relevant words, such as "*close*" and "*very convenient*" for aspect *location*, while "*a little uncomfortable*" and "*nitpicking*"for *room*.

Error Type Breakdown. We analyze error cases in the experiments and broadly categorize them into three types: **(1)** The first type of errors are due to negation words. For instance, for the review "*The taste of this beer is not good, don't buy it*", HRL could precisely select the sentiment


Clause1 [This hotel is close to railway station]
Clause2 [and very convenient to eat around]
Clause3 [but room of Hilton is a little uncomfortable .]
Clause4 [I am often nitpicking for room decoration .]
Clause5 [Besides , the price is very expensive]
Clause6 [although the staff service is professional .]
Clause1 [This hotel is close to railway station]
Clause2 [and very convenient to eat around]
Clause3 [but room of Hilton is a little uncomfortable .]
Clause4 [I am often nitpicking for room decoration .]
Clause5 [Besides , the price is very expensive]
Clause6 [although the staff service is professional .]

(a) For aspect ***location*** (b) For aspect ***room***

Figure 4: Visualizations of clause selection (along with the row ↓) and word selection (along with the column →) for aspect (a) ***location*** and (b) ***room***. Red denotes the clause has been selected, blue denotes the word has been selected and other colors denote the token has been discarded.

"*good*", but fail to select the negation word "*not*". This inspires us to work on optimizing our approach in order to capture negation scope better in our future work. **(2)** The second type of errors are due to comparative opinions. For instance, for the review "*The room of Sheraton is much better than this one.*", HRL incorrectly predicts high rating (5 stars) to aspect *room*. It would be interesting to see if incorporating syntactic information can solve this problem and bring performance improvement. **(3)** Finally, some errors are due to mistakes during clause splitting (i.e., EDU splitting). For instance, for the review "[*This hotel having good location*] [*often needs lots of time to check in.*]", it is assigned into one clause while it talks two aspects, i.e., *location* and *check in/front desk*. This encourages to improve the performance of clause splitting for informal review texts.

5 Related Work

Aspect Sentiment Classification. Traditional studies for DASC mainly focus on feature engineering to explore efficient features for DASC (Titov and McDonald, 2008; Lu et al., 2011; McAuley et al., 2012). Recently, neural networks with the characteristic of automatically mining features have shown promising results on DASC. Lei et al. (2016) focused on extracting rationales for aspects and build a neural text regressor to predict aspect rating; Yin et al. (2017) focused on using hierarchical iterative attention to learn aspect-specific text representation for DASC; Li et al. (2018) employed a hierarchical attention approach to DASC which incorporates both the external user and overall rating information. Besides, neural networks have been widely adopted for performing a closely related task, i.e., Sentence-level Aspect Sentiment Classification (Wang et al., 2016; Tang et al., 2016; Wang and Lu, 2018).

Reinforcement Learning. In recent years, reinforcement learning has been applied successfully to some NLP tasks. Guo (2015) employed deep Q-learning to improve the seq2seq model for the text generation task; Li et al. (2016) showed how to apply deep reinforcement learning to model future reward in the chatbot dialogue task; Takanobu et al. (2018) employed hierarchical reinforcement learning to model the relation extraction task; Zhang et al. (2018) combined LSTM with reinforcement learning to learn structured representations for the text classification task, which is inspirational to our approach.

Unlike all above studies, inspired by the cognitive process of human beings, this paper proposes a new HRL approach to DASC task. To the best of our knowledge, this is the first attempt to address DASC with HRL.

6 Conclusion

In this paper, we propose a hierarchical reinforcement learning approach to DASC. The main idea of the proposed approach is to perform sentiment classification like human beings. Specifically, our approach employs a high-level policy and a low-level policy to perform clause selection and word selection in DASC respectively. Experimentation shows that both the clause and word selection are effective for DASC and the proposed approach significantly outperforms several state-of-the-art baselines for DASC.

In our future work, we would like to solve other challenges in DASC, e.g., negation detection problem, to further improve the performance. Furthermore, we would like to apply our HRL approach to other sentiment analysis tasks, such as aspect and opinion co-extraction, and dialog-level sentiment analysis.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by three NSFC grants, i.e., No.61672366, No.61702149 and No.61525205. This work was also supported by the joint research project of Alibaba Group and Soochow University, and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

A. Bayoudhi, H. Ghorbel, H. Koubaa, and L. Belguith. 2015. Sentiment classification at discourse segment level: Experiments on multi-domain arabic corpus. *JLCL*, 30(1).

Hongyu Guo. 2015. Generating text with deep reinforcement learning. *CoRR*, abs/1510.09202.

D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

T. Lei, R. Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP*.

J. Li, H. Yang, and C. Zong. 2018. Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *Proceedings of COLING*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *CoRR*, abs/1606.01541.

S. Li, C. Huang, G. Zhou, and Sophia Yat Mei Lee. 2010. Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Proceedings of ACL*.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proceedings of ICLR*.

X. Liu, L. Mou, H. Cui, Z. Lu, and S. Song. 2018. JUMPER: learning when to make classification decisions in reading. *CoRR*, abs/1807.02314.

B. Lu, M. Ott, C. Cardie, and B. K. Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *Proceedings of ICDM*.

W. MANN. 1988. Rhetorical structure theory : Toward a functional theory of text organization. *Text*, 8(3):243–281.

G. Marcus. 2018. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631.

J. McAuley, J. Leskovec, and D. Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of ICDM*.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining*, page 5. ACM.

B. Pang and L. Lee. 2007. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2).

R. Sutton, D. McAllester, and etc. 1999a. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*.

R. Sutton, D. Precup, and S. Singh. 1999b. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. *CoRR*, abs/1811.03925.

D. Tang, B. Qin, and T. Li. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of EMNLP*.

D. Tang, B. Qin, and T. Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP*.

I. Titov and R. McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL*.

B. Wang and W. Lu. 2018. Learning latent opinions for aspect-level sentiment classification. In *Proceedings of AAAI*.

H. Wang, Y. Lu, and C. Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of KDD*.

J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, and G. Zhou. 2018. Aspect sentiment classification with both word-level and clause-level attention networks. In *Proceedings of IJCAI*.

Y. Wang, M. Huang, X. Zhu, and L. Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of EMNLP*.

R. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8.

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR*.

Y. Yin, Y. Song, and M. Zhang. 2017. Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of EMNLP*.

T. Zhang, M. Huang, and L. Zhao. 2018. Learning structured representation for text classification via reinforcement learning. In *Proceedings of AAAI*.

Y. Zou, T. Gui, Q. Zhang, and X. Huang. 2018. A lexicon-based supervised attention model for neural sentiment analysis. In *Proceedings of COLING*.