A reinforcement learning paradigm for Vietnamese aspect-based sentiment analysis

Viet The Bui[1], Linh Thuy Ngo[2,3], Oanh Thi Tran[2]
[1]School of Computing and Information Systems, Singapore Management University, Singapore, Singapore
[2]International School, Vietnam National University Hanoi, Hanoi, Vietnam
[3]Banking Academy of Vietnam, Hanoi, Vietnam

Article Info

Article history:

Received Sep 3, 2024
Revised Feb 19, 2025
Accepted Mar 15, 2025

Keywords:

Aspect-based sentiment analysis
Deep learning
Pre-trained models
Reinforcement learning
Vietnamese

ABSTRACT

This paper presents the task of aspect-based sentiment analysis (ABSA) that recognizes the sentiment polarity associated with each aspect of entities discussed in customers' reviews, focusing on a low-resourced language, Vietnamese. Unlike conventional classification approaches, we leverage reinforcement learning (RL) techniques by formulating the task as a Markov decision process. This approach allows an RL agent to handle the hierarchical nature of ABSA, sequentially predicting entities, aspects, and sentiments by exploiting review features and previously predicted labels. The agent seeks to discover optimal policies by maximizing cumulative long-term rewards through accurate entity, aspect, and sentiment predictions. The experimental results on public Vietnamese datasets showed that the proposed approach yielded new state of the art (SOTA) results in both hotel and restaurant domains. Using the best model, we achieved an improvement of 1% to 3% in the F1 scores for detecting aspects and the corresponding sentiment polarity.

Corresponding Author:

Oanh Thi Tran
International School, Vietnam National University Hanoi
144 XuanThuy, CauGiay, Hanoi, Vietnam
Email: tranthioanh@vnu.edu.vn

1. INTRODUCTION

Aspect-based sentiment analysis (ABSA) involves identifying sentiments directed towards specific aspects of entities mentioned in texts. It has been widely applied in monitoring brand reputation, analyzing customer feedback, and tracking public sentiment towards social and political events. Formally, given a text, a prediction model is built to determine the set of triples $< e, a, p >$, where e represents entities, a denotes the associated aspects, and p indicates the sentiment polarity. Figure 1 illustrates a customer review of four key triples which are ROOM, CLEANLINESS, positive; SERVICE, GENERAL, positive; FOOD&DRINK, QUALITY, positive; and FOOD&DRINK, STYLE&OPTION, negative.

The structure of ABSA prediction is inherently hierarchical, reflecting the natural progression of sentiment expression: first, entities are identified; next, aspects related to each entity are extracted; and finally, the sentiment associated with each aspect is determined. This order - entities $\rightarrow$ aspects $\rightarrow$ polarity enables each prediction stage to leverage the context established in the preceding stages, creating a more accurate and context-sensitive sentiment analysis. Until now, conventional approaches have framed the task as a classification problem, using independent or joint learning methods. Researchers have explored incorporating deep learning methods, such as recurrent neural networks [1], [2], convolution neural networks [3], transformer models [4]-[7], attention models [8], and multi-task solutions [9] to capture the contextual and semantic nuances of input texts for sentiment analysis. These techniques have shown promising results in capturing fine-grained sentiment information, unfortunately, they do not well capture the context-dependent dependencies among entities, aspects, and polarity.

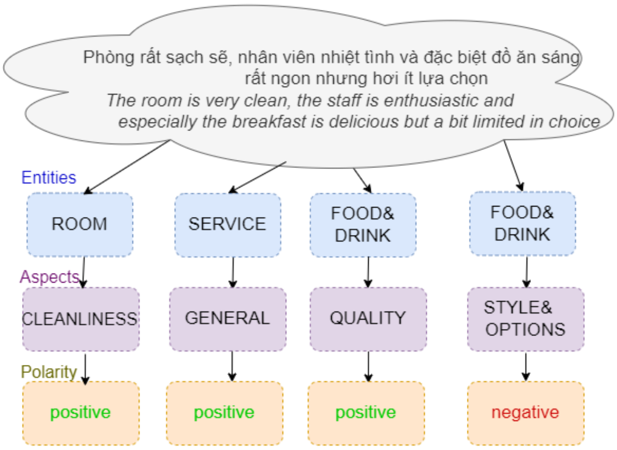

Phòng rất sạch sẽ, nhân viên nhiệt tình và đặc biệt đồ ăn sáng rất ngon nhưng hơi ít lựa chọn
The room is very clean, the staff is enthusiastic and especially the breakfast is delicious but a bit limited in choice
Entities
ROOM
SERVICE
FOOD& DRINK
FOOD& DRINK
Aspects
CLEANLINESS
GENERAL
QUALITY
STYLE& OPTIONS
Polarity
positive
positive
positive
negative

Figure 1. A sample review's hierarchical structure, annotated with entities, their aspects, and polarity

Recently, reinforcement learning (RL) [10] has shown promise in various domains such as robotics [11], wireless networks [12], and game [13]. However, its application to natural language processing (NLP) tasks remains largely restricted to conversational systems [14]. Given RL 's unique strength in handling sequential decision-making, it presents a promising approach for ABSA. By treating ABSA as a Markov decision process, ABSA's hierarchical structure - progressing from entity recognition to aspect identification and, finally, to sentiment prediction - can be managed as a series of sequential decisions. In this setting, deep Q-learning is leveraged to dynamically update state representations as it identifies entities, assigns them to relevant aspects, and finally predicts the sentiment for each aspect. Through optimizing these decisions for long-term rewards, the approach is expected to improve classification accuracy by capturing the structured dependencies often missed in traditional classification methods. To our knowledge, no prior work has framed ABSA as an RL problem, especially for low-resource languages, making this a novel direction in sentiment analysis. This paper makes the following contributions: i) introduces a new approach using RL to solve the ABSA task with a specific focus on a low-resource language, a.k.a Vietnamese; and ii) extensively conducts different experiments to verify the effectiveness of the proposed approach on two available benchmark datasets.

The remainder of the paper is structured as follows. Section 2 describes related work. Section 3 shows how to treat ABSA as a Markov decision process. Then, section 4 mathematically designs an RL agent to solve this Markov decision process. Section 5 presents the deep Q-learning to solve the ABSA in the RL setting. Section 6 describes the public benchmark corpus, setups of experiments, experimental results, and some discussion. Finally, we conclude the paper and point out some future lines of work in section 7.

2. RELATED WORK

Research on sentiment analysis has been performed for different levels such as document level, sentence level, phrase level, and aspect level [15]. There are two mainly used approaches to explore sentiments: lexicon-based approaches and machine learning approaches. The former one [16] is extremely feasible for sentiment analysis at the sentence and phrase levels. The latter one enables systems to acquire new abilities without being explicitly programmed to do so. Some typical work is a naive Bayes model along with an support vector machine (SVM) model [17], semi-supervised machine learning that integrates pre-processing and classification algorithms for unlabelled datasets [18], attention (CNN-RNN) [8]. In recent years, sentiment analysis has become more prominent, and ASBA is a valuable and quickly expanding research field. Complex algorithms [3], [5], [7], [9], [19] like long short-term memory (LSTM), pre-trained models like BERT, and GPT-2 may be used to accomplish the task. These classification techniques have shown promising results in capturing fine-grained sentiment information. Unfortunately, they may not effectively capture the hierarchical and interdependent nature of aspect-based sentiments.

RL [10] is a powerful machine learning technique where an agent learns to make decisions by interacting with an environment, receiving rewards or punishments, and adjusting its actions to maximize long-term cumulative rewards. It is primarily used for sequential decision-making tasks rather than classification tasks. For example, several RL models have been studied in planning and control problems [20], action-conditional video prediction [21], and wireless networks [12]. RL can be combined with other techniques, such as deep learning, to perform tasks in NLP such as sequence-to-sequence learning for text generation and classification tasks [14], [22]. RL demonstrates promise in NLP classification tasks, especially those with multiple or hierarchical labels, where capturing the inter-dependencies among various labels is essential. Therefore, incorporating RL can be an interesting research direction to explore and can potentially lead to novel approaches and improved ABSA models.

3. ABSA AS A MARKOV DECISION PROCESS

We suppose a review x is associated with n label triples of (a_e^i, a_a^i, a_p^i) $(i = 1...n)$ which involve n sentiment polarity a_p^i towards targeted entities a_e^i and their associated aspects a_a^i. This can be modeled as a step-by-step decision-making process as shown in Figure 2:

i) Step 1. Initialization: start with all states initialized with the input text x where no entities, aspects, or sentiments are identified, and x is yet to be processed.

ii) Step 2. Sequential decision-making for the i^{th} triple

- *Entity identification*: at the state s^{t+1}, the agent scans x and identifies the i^{th} entity a_e^i (e.g., 'hotel'). Update the state to s^{t+2} to include a_e^i, where $t = (i-1) * 4$.
- *Aspect identification*: based on the current state s^{t+2}, the agent identifies an aspect a_a^i related to the entity a_e^i (e.g., 'cleanliness' of 'hotel'). Update the state to s^{t+3} to include a_e^i, a_a^i.
- *Sentiment prediction*: the agent predicts the sentiment a_p^i for the identified a_a^i (e.g., *positive* sentiment for 'cleanliness'). Update the state to s^{t+4} to include a_e^i, a_a^i, a_p^i.

iii) Step 3. Decision on next triple prediction: the agent moves to the next part of the text to identify more entities, aspects, and sentiments by repeating Step 2 - Sequential decision-making for the $(i+1)^{th}$ triple. It updates the next state for the next round of the sequential decision-making process until the entire text is processed. Otherwise, it reaches the ending state $s^{n\times4}$.

iv) Step 4. Reward assignment: the agent receives rewards based on the accuracy of its predictions. Cumulative rewards guide the learning process, encouraging the agent to refine its policy for better future predictions.

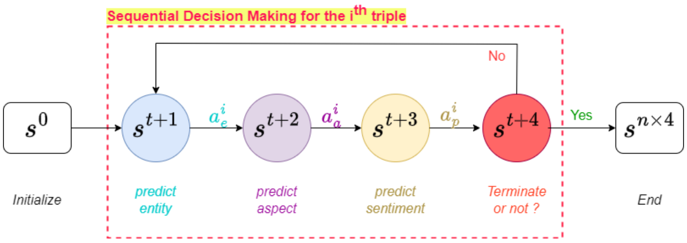


Figure 2. The visualization of ABSA as a Markov decision process

4. MATHEMATICALLY DESIGN AN RL AGENT TO SOLVE ABSA

The RL agent is represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathbf{Y}, \mathbf{q}, \mathbf{r}, \gamma)$, where $\mathcal{S}$ refers to the set of states, $\mathcal{A}$ denotes the set of finite possible actions (i.e., $|\mathcal{A}| < \infty$), $\mathbf{Y}$ is the ground truth labels, $\mathbf{q}$ is transition probabilities by taking an action in a given state, $\mathbf{r}$ is reward function, and $\gamma \in [0, 1]$ is a discount factor. A deterministic curricular policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is a mapping from states to actions as follows:

i) State: the state $s \sim \mathcal{S}$ is represented as a tuple (f, h), where f is a vector representation of the review x from input dataset $\mathcal{D}$; and h is an action history. In this study, x is represented by a list of consecutive token indices tokenized by a sub-word tokenizer (see Figure 3). This tokenizer is based on SentencePiece with a certain number of sub-word token strings, converts tokens strings to ids and back, and adds several special token strings to format the input review such as: [CLS] - add to the begin of the sentence to represent the whole tokens; [UNK] - represent unknown tokens which are not tokenized by tokenizer with the given vocabulary; and [PAD] - extend padding tokens to the list of token strings to make all input reviews have the same length. Review f is forwarded into a pre-trained transformer-based model $\mathcal{B}_\theta$ (e.g., BERT, ELECTRA, etc.), where θ refers to trained parameters, for extracting features. After that, we use the output hidden vector of token [CLS] as a representation vector $H(f, \theta) = \mathcal{B}_\theta(f)_{[CLS]} \in \mathbb{R}^{d_{\mathcal{B}_\theta}}$, where θ denotes pre-trained parameters, and $d_{\mathcal{B}_\theta}$ is the hidden output dimension of $\mathcal{B}_\theta$.

ii) Action: the agent learns to predict a set of tuple of entity-aspect-polarity action $A = \{(a^i_{\mathbf{e}}, a^i_{\mathbf{a}}, a^i_{\mathbf{p}})\}_{i=1..n}$, where $a^i_{\mathbf{e}} \in \mathcal{A}_{\mathbf{e}}$, $a^i_{\mathbf{a}} \in \mathcal{A}_{\mathbf{a}}$, and $a^i_{\mathbf{p}} \in \mathcal{A}_{\mathbf{p}}$ are entity, aspect, and polarity actions respectively; n is the number of predicted actions. Due to the hierarchical structure of target labels, each action is predicted by the current state s and action history h. Therefore, given by a policy model π, sequentially we predict an entity first, an aspect next, and a polarity last as shown in Figure 2.

$$a_{\mathbf{e}} \sim \pi(H_s, h|a_{\mathbf{e}}); h' \leftarrow h \cup \{a_{\mathbf{e}}\}$$

$$a_{\mathbf{a}} \sim \pi(H_s, h'|a_{\mathbf{a}}); h'' \leftarrow h' \cup \{a_{\mathbf{a}}\}$$

$$a_{\mathbf{p}} \sim \pi(H_s, h''|a_{\mathbf{p}}); h''' \leftarrow h'' \cup \{a_{\mathbf{p}}\}$$

We also add another binary step $a_{\mathbf{t}} \in \mathcal{A}_{\mathbf{t}} = \{0, 1\}$ (**t** stands for terminate) after predicting each tuple action $(a_{\mathbf{e}}, a_{\mathbf{a}}, a_{\mathbf{p}})$ to detect whether this is the last item of A or not. We define $a_{\mathbf{t}}$ as 1 if the agent wants to predict another tuple action, and as 0 otherwise. Basically, our policy aims to predict consequence tuples $(a^i_{\mathbf{e}}, a^i_{\mathbf{a}}, a^i_{\mathbf{p}}, a^i_{\mathbf{t}})$ for solving hierarchical multi-label ABSA classification:

$$(a^1_{\mathbf{e}}, a^1_{\mathbf{a}}, a^1_{\mathbf{p}}, a^1_{\mathbf{t}}), (a^2_{\mathbf{e}}, a^2_{\mathbf{a}}, a^2_{\mathbf{p}}, a^2_{\mathbf{t}}), ..., (a^n_{\mathbf{e}}, a^n_{\mathbf{a}}, a^n_{\mathbf{p}}, a^n_{\mathbf{t}})$$

where n is the number of predicted triples.

iii) Transitions: as described by the action prediction process, each episode (trajectory) has $4 \times n$ steps. Each step is represented as a tuple (f, h), and after taking an action a, the next state becomes (f, h'), where $h' = h \cup \{a\}$.

iv) Reward: we combine two types of rewards into one reward r_t at step t:

$$r_t = \delta_t \mathbb{I}[a^t_{\mathbf{t}} == 1] r_{1,t} + (1 - \delta_t) \mathbb{I}[T == 0] r_{2,t}$$

Where $r_{1,t}$ and $r_{2,t}$ denotes dense exploration and sparse competition rewards respectively; δ_t is a linear annealing factor. Our target is finding an optimal policy π that maximizes the discounted long-term reward, which means the higher reward is the better score. The dense exploration reward $r_{1,t}$ is calculated by the number of correct labels and the sparse competition reward $r_{2,t}$ is calculated at the end of each episode τ by using a F_1-score metric with a pair of predicted actions A_τ and its targets Y_τ:

$$r_{2,t} = F_1(A_\tau, Y_\tau) = \frac{2}{Pre^{-1}(A_\tau, Y_\tau) + Rec^{-1}(A_\tau, Y_\tau)} \in \mathbb{R}$$

where Pre denotes precision score and Rec denotes recall score.

In summary, we treat ABSA problem as an Markov decision process task where the agent learns to predict a list of consecutive actions and maximize the reward represented by F_1-score. If the number of transitions is too small or too large, the FP and FN might be much bigger than TP, the reward will be smaller, and the agent could be worse. This environment is challenging to make the F_1-score higher for every incremental timestep.

5. DEEP Q-LEARNING FOR ABSA

In this section, we delve into the architecture and methodology of our deep Q-network (DQN) model, which is tailored to address the complex requirements of the tasks. As depicted in Figure 3, the architecture begins by processing the input review through a pre-trained transformer model, such as BERT or ELECTRA, to extract a comprehensive and contextualized representation of the review. This representation serves as the input for the DQN, which acts as an agent learning to predict a sequence of actions necessary for navigating the hierarchical structure of the ABSA task. During the training phase, the primary focus is on refining the parameters of the Q-network, ensuring that it can effectively learn from the data and improve its predictive capabilities.

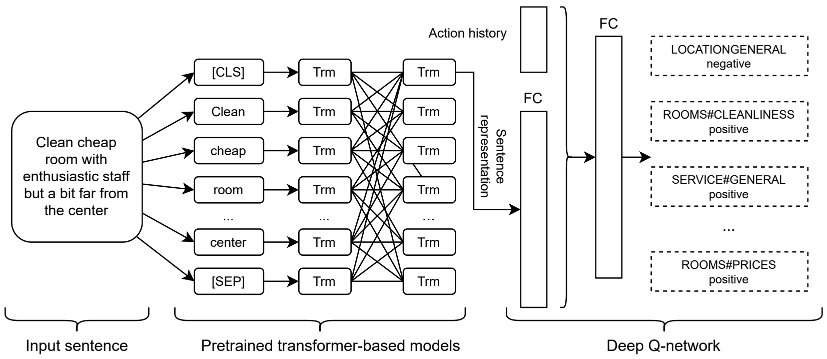


Figure 3. An overview of our model architecture using pre-trained transformer-based models as review representation encoders and DQN as an agent to learn to predict a sequence of actions (labels) for solving multi-label text classification

In the conventional RL models, the agent selects one action from the entire action set at each step, often without constraints, which can lead to repetitive actions within an episode. However, in the context of ABSA, the label triples $< e, a, p >$ for a given review must be unique, preventing duplication within an episode. To adapt this requirement to the RL framework, our model is designed to ensure that the agent takes non-repetitive actions throughout both the training and testing phases. We manage duplication by employing a three-level tree structure. The initial level comprises the root, succeeded by entities, their respective aspects, and ending with their potential polarity. Traversing from the root to the leaf will establish the triples of $< e, a, p >$. Any previously visited branch is blocked by setting the log-likelihood to $-inf$, equating to a probability of 0. To maintain sequential labeling, we update the action set after each step to remove duplication and enforce the agent to predict entities first, followed by aspects, and finally their polarity.

In a deterministic environment, we simplify the problem by assuming a fixed relationship between actions and subsequent states. The goal is to train a policy π that maximizes the expected discounted cumulative reward, also known as the *return*, from any starting state s_0. This is mathematically expressed as:

$$R_{t_0} = \sum_{t=t_0}^{\infty} \gamma^{t-t_0} r_t$$

Where $\gamma \in [0, 1)$ is the discount factor. The discount factor γ ensures that the infinite sum converges by reducing the weight of future rewards, thereby prioritizing immediate rewards over distant ones. This is crucial in balancing the trade-off between short-term and long-term rewards.

The core of DQN is the action-value function $Q^* : S \times A \to \mathbb{R}$, which provides the expected return of taking action a in state s and following the optimal policy thereafter. The optimal policy π^* can be derived by selecting the action that maximizes this function:

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

Since the true Q^* is unknown, we use a neural network to approximate it. The network parameters θ are updated to minimize the difference between the predicted Q-values and the target Q-values, derived from the Bellman equation:

$$Q^\pi(s, a) = \mathbb{E}\left[r + \gamma Q^\pi(s', \pi(s')) \mid s, a\right]$$

For the optimal Q-function Q^*, the Bellman optimality equation is:

$$Q^*(s,a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right]$$

In practice, the Q-learning update rule is applied to iteratively adjust the parameters θ by minimizing the temporal difference error. Instead of using mean squared error (MSE), which is sensitive to outliers, we use the Huber loss. The Huber loss is defined as:

$$L(\theta) = \mathbb{E}_{(s,a,s',r)\in\mathcal{D}}\left[\mathcal{L}_\theta(\delta)\right]$$

where $\mathcal{D}$ is the replay buffer containing past experiences, and the loss function $\mathcal{L}_\theta(\delta)$ is:

$$\mathcal{L}_\theta(\delta) = \begin{cases} \frac{1}{2}\delta^2(\theta) & \text{if } |\delta(\theta)| \leq 1 \\ |\delta(\theta)| - \frac{1}{2} & \text{otherwise} \end{cases}$$

The Huber loss behaves quadratically for small errors (like MSE) and linearly for large errors (like mean absolute error (MAE)), providing robustness to outliers in the Q-value estimates. The update to the parameters θ during training is performed using stochastic gradient descent or its variants, such as Adam [23], by computing the gradient of the loss function with respect to θ:

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$$

Where α is the learning rate, controlling the step size of the update. This detailed formulation allows the DQN to effectively learn an approximation of the optimal Q-function by iteratively minimizing the temporal difference error, thereby improving the policy π to maximize the expected return.

The deep Q-learning algorithm for reinforced ABSA is detailed in Algorithm 1. Our model employs a feed-forward neural network designed to handle state transitions by taking as input the difference between the current state s_t and the previous state s_{t-1}, represented as Δs_t.

Algorithm 1 DQN for ABSA

```
Input: Environment $env$; Q network $Q_\theta$; trainable parameters $\theta$; learning rate $\alpha$; replay buffer $\mathcal{D}$.
$\theta' \leftarrow \theta$                                          ▷ Initialize V parameters
while not coverage do
    $s_t \leftarrow env.reset()$                                   ▷ Reset environment
    for each environment step do
        $a_t \leftarrow \arg\max_{a_t} Q_\theta(s_t, a_t)$                  ▷ Sample action
        $s_{t+1} \leftarrow env.step(a_t)$                             ▷ Take an action $a_t$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, R(a_t|s_t), s_{t+1})\}$    ▷ Append transition to buffer $\mathcal{D}$
        $s_t \leftarrow s_{t+1}$
    end for
    for each gradient step do
        $\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$                  ▷ Update Q network
        $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
    end for
end while
```

The neural network outputs a vector of Q-values, $Q(s,a;\theta)$, where θ represents the parameters of the neural network. The number of outputs corresponds to the discrete action space, which has a total dimension given by:

$$|\mathcal{A}| = |\mathcal{A}_{\text{entity}}| + |\mathcal{A}_{\text{aspect}}| + |\mathcal{A}_{\text{polarity}}| + |\mathcal{A}_{\text{termination}}|$$

Where $|\mathcal{A}_{\text{entity}}|$, $|\mathcal{A}_{\text{aspect}}|$, $|\mathcal{A}_{\text{polarity}}|$, and $|\mathcal{A}_{\text{termination}}|$ are the sizes of the action sets for entities, aspects, polarity, and termination, respectively.

The Q-values are computed as $Q(s, a; \theta) = f(\Delta s_t; \theta)$, where f is the function approximated by the neural network, mapping state differences to Q-values for each possible action a. To extract meaningful features from each input review, we utilize a pre-trained model $\mathcal{B}_\theta$ as the feature representation model. This model processes the raw input x of a review and generates a feature vector $\phi(x)$:

$$\phi(x) = \mathcal{B}_\theta(x)$$

This feature vector $\phi(x)$ serves as the input to the neural network, allowing the Q-learning algorithm to leverage pre-trained knowledge for improved performance in sentiment analysis tasks. The action selection in each state involves choosing the action a^* that maximizes the Q-value:

$$a^* = \arg\max_{a \in \mathcal{A}} Q(s, a; \theta)$$

This action selection strategy ensures that the policy π is aimed at maximizing the expected cumulative reward over time. By integrating these mathematical formulations, the model effectively learns to map state-action pairs to their expected returns, thereby optimizing the policy for sentiment analysis tasks in a discrete action space.

6. EXPERIMENTS

This section first presents the dataset used to conduct experiments. Then, experimental setups such as evaluation metrics and model training parameters are shown. Finally, experimental results are elaborated and compared with several strong SOTA baselines.

6.1. Datasets

The dataset used to perform experiments is a public dataset released in the ABSA challenges of VLSP 2018 at https://vlsp.org.vn/vlsp2018/eval/sa. The campaign dealt with data in two domains, namely restaurants and hotels. The number of entities, their aspects, and corresponding polarity mentioned in each review are diverse. The information about the training, development, and test sets is summarized in Table 1. For each review, its aspects are identified by the tuples (entities, attributes) (Phase A). While in Phase B, we need to further identify the polarity for each aspect tuple as positive, negative, or neutral.

Table 1. Statistics on the train, dev, and test sets of two datasets on two domains

Domain	Dataset	#Reviews	#Aspects
Restaurant	Training	2961	9034
	Validation	1290	3408
	Test	500	2419
Hotel	Training	3000	13948
	Validation	2000	7111
	Test	600	2584

6.2. Experimental setups

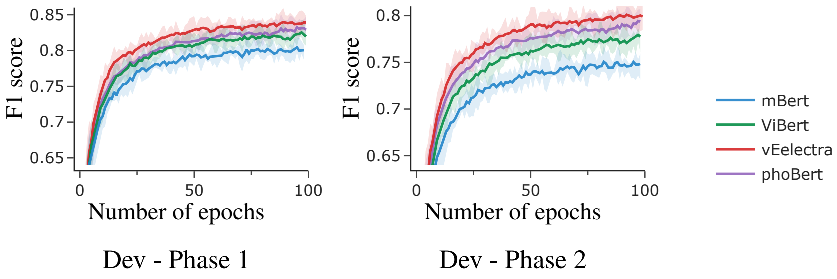
To evaluate the performance, we use the metrics proposed by the campaign. They are precision, recall, and F1-score in micro-averaged. In conducting experiments, we implemented the models using Pytorch. For pre-training language models, we exploited different ones available for Vietnamese. They are mBERT [24], viBERT [25], phoBERT [26], and vElectra [25]. The models' hyper-parameters were chosen via a search on the validation set. We varied different hyper-parameters to find the optimized sets for ones such as filter window sizes, dropout rates, optimization methods, learning rates, batch sizes, and number of epochs.

6.3. Experimental results

Two kinds of experiments were implemented. First, we conducted experiments to verify how pre-trained transformer-based models impact the prediction performance of the proposed RL-based approach. Then, we compare this proposed approach with the existing SOTA results on two benchmark datasets.

6.3.1. The effectiveness of different pre-trained language models on the RL-based approach

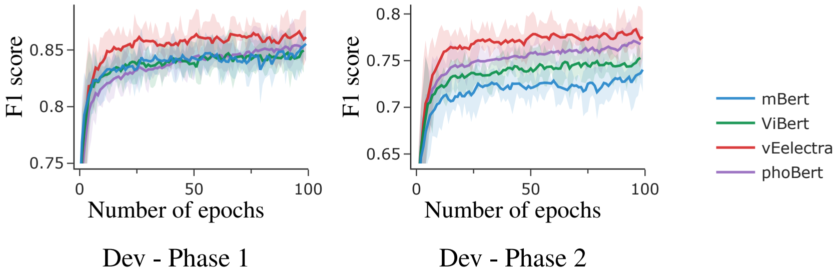
Figures 4 and 5 draw the convergence curves for the RL approach using four available pre-trained language models on two phases. They illustrate the convergence patterns observed in the hotel and restaurant datasets. It's evident from these plots that the algorithms converge effectively after several iterations on development sets. Notably, employing vElectra as a review encoder resulted in the highest performance, followed by phoBert, viBERT, and mBert.



Dev - Phase 1

Dev - Phase 2

Figure 4. Convergence curves on the development set of the hotel dataset



Dev - Phase 1

Dev - Phase 2

Figure 5. Convergence curves on the development set of the restaurant dataset

6.3.2. Comparing the RL-based approach with the existing SOTA work

This section presents the experimental results of the proposed approach compared to the SOTA results on two public Vietnamese datasets. To conduct a comprehensive comparison, we've chosen the best three existing methods: JointLearning-mBERT, JointLearning-viBERT, and MultiTask-PhoBERT. For the RL approach, we implemented the algorithm using four different backbone models. Each model utilizes a distinct pre-trained language model for Vietnamese: mBERT, viBERT, phoBERT, and vElectra. Tables 2 and 3 show experimental results on the test sets of two domains.

For the hotel domain, it can be seen from Table 2 that the RL approach performed better than all existing models. Using vElectra and phoBERT yielded the best performance. Notably, vElectra slightly captures the nuances of reviews better than phoBERT, leading to improved F1 scores in both phases. We achieve a new SOTA of 83.51% in the F1 score for Phase 1, and 78.55% in the F1 score for Phase 2. Using the best RL-vElectra model, we achieved an F1 score of 83.52% for Phase 1 and 78.55% for Phase 2 on the hotel dataset.

For the restaurant domain, the RL approach demonstrated notably superior performance compared to the three existing SOTA models (see Table 3). We also see an improvement in the F1 scores in both two phases. Compared to the best model of MultiTask-phoBERT, the RL-vElectra significantly boosted the F1 scores by 1.02% for Phase 1, and 2.8% for Phase 2. Using the best RL-vElectra yielded the highest performance of 85.25% for Phase 1, and 74.35% for Phase 2 on the restaurant dataset.

Using the same pre-trained language models, the proposed RL approach consistently outperformed the conventional classification approach across both datasets. Notably higher performance was achieved with both mBERT and viBERT, while phoBERT yielded slightly better results. Among all language models, using v-Electra achieving the best performance and established a new SOTA result on both two datasets. These findings demonstrate that the proposed RL approach surpasses the existing SOTA classification methods.

Table 2. Experimental results on the test set of the hotel domain

		Phase 1			Phase 2	
	Pre	Rec	F1	Pre	Rec	F1
Baselines						
JointLearning-mBert [25]	85.32	76.04	80.42	78.17	65.56	71.31
JointLearning-viBert [25]	83.93	80.26	82.06	80.04	70.01	74.69
MultiTask-phoBERT [9]	87.45	78.17	82.55	81.90	73.22	77.32
Reinforcement Learning						
RL-mBert	85.96	77.38	81.45	79.67	71.13	75.16
RL-viBert	86.59	79.83	83.08	81.04	74.30	77.52
RL-phoBert	86.28	79.53	82.76	81.38	74.65	77.87
RL-vElectra	87.82	79.60	83.51	82.86	74.66	78.55

Table 3. Experimental results on the test set of the restaurant domain

		Phase 1			Phase 2	
	Pre	Rec	F1	Pre	Rec	F1
Baselines						
JointLearning-mBert [25]	84.03	82.64	83.33	65.70	71.43	68.45
JointLearning-viBert [25]	84.21	84.25	84.23	69.75	72.92	71.30
MultiTask-phoBERT [9]	-	-	84.23	69.66	73.54	71.55
Reinforcement Learning						
RL-mBert	83.57	84.61	84.09	69.54	70.58	70.05
RL-viBert	84.35	84.46	84.41	72.73	72.84	72.78
RL-phoBert	84.13	84.46	84.30	72.86	73.19	73.02
RL-vElectra	85.89	84.62	85.25	74.99	73.72	74.35

7. CONCLUSION

In this paper, we present a new and effective approach to sentiment analysis for Vietnamese which leverages RL techniques instead of conventional approaches. Instead of modelling the tasks as a classification problem with supervised learning techniques, we formulate the prediction task as a Markov decision process. Here, we deploy a RL agent to sequentially predict labels, effectively leveraging review features and previously predicted labels. The agent seeks to discover optimal policies by maximizing long-term rewards, reflecting prediction accuracy. The experimental results on public Vietnamese datasets showed that the proposed approach yielded new SOTA results in both two domains of hotels and restaurants. Specifically, we achieved 83.51% in the F1 score for Phase 1, and 78.55% in the F1 score for Phase 2 on the hotel dataset. Employing the identical RL-vElectra model likewise achieved the highest performance, reaching 85.25% for Phase 1 and 74.35% for Phase 2 on the restaurant dataset. In the future, we continue to try different types of reward functions to find the best one. Other kinds of large language models should also be integrated to optimize the review representation that captures the full context of reviews.

FUNDING INFORMATION

This paper is funded by the International School, Vietnam National University Hanoi (VNU-IS) under the local project CS.2023-11.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contri butions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Viet The Bui	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Linh Thuy Ngo		✓				✓		✓	✓	✓	✓	✓		
Oanh Thi Tran	✓		✓	✓		✓			✓	✓	✓		✓	✓

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the competition VLSP 2018 - Aspect Based Sentiment Analysis. Data are available at https://vlsp.org.vn/vlsp2018/eval/sa with the permission of VLSP representatives.

REFERENCES

[1] D. V. Thin, V. D. Nguye, K. V. Nguyen, and N. L.-T. Nguyen, "Deep learning for aspect detection on Vietnamese reviews," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, IEEE, Nov. 2018, pp. 104–109, doi: 10.1109/NICS.2018.8606857.

[2] H.-Q. Nguyen and Q.-U. Nguyen, "An ensemble of shallow and deep learning algorithms for Vietnamese sentiment analysis," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, IEEE, Nov. 2018, pp. 165–170, doi: 10.1109/NICS.2018.8606880.

[3] J. Meng, Y. Long, Y. Yu, D. Zhao, and S. Liu, "Cross-domain text sentiment analysis based on CNN_FT method," *Information*, vol. 10, no. 5, May 2019, doi: 10.3390/info10050162.

[4] N. C. Le, N. T. Lam, S. H. Nguyen, and D. T. Nguyen, "On Vietnamese sentiment analysis: a transfer learning method," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, IEEE, Oct. 2020, pp. 1–5, doi: 10.1109/RIVF48685.2020.9140757.

[5] M. Singh, A. K. Jakhar, and S. Pandey, "Sentiment analysis on the impact of coronavirus in social life using the BERT model," *Social Network Analysis and Mining*, vol. 11, no. 1, Dec. 2021, doi: 10.1007/s13278-021-00737-z.

[6] O. T. Tran and V. T. Bui, "A BERT-based Hierarchical model for Vietnamese aspect based sentiment analysis," in *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, IEEE, Nov. 2020, pp. 269–274, doi: 10.1109/KSE50997.2020.9287650.

[7] M. Zouidine and M. Khalil, "Arabic sentiment analysis based on deep reinforcement learning," in *2022 5th International Conference on Networking, Information Systems and Security: Envisage Intelligent Systems in 5g//6G-based Interconnected Digital Worlds (NISS)*, IEEE, Mar. 2022, pp. 1–5, doi: 10.1109/NISS55057.2022.10085147.

[8] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, Feb. 2021, doi: 10.1016/j.future.2020.08.005.

[9] H.-Q. Dang, D.-D.-A. Nguyen, and T.-H. Do, "Multi-task solution for aspect category sentiment analysis on Vietnamese datasets," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, IEEE, Jun. 2022, pp. 404–409, doi: 10.1109/CyberneticsCom55287.2022.9865479.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Cambridge, Massachusetts: MIT Press, 1998.

[11] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic Helicopter flight," in *NIPS 2006: Proceedings of the 19th International Conference on Neural Information Processing Systems*, in NIPS'06. Cambridge, MA, USA: MIT Press, 2006, pp. 1–8, doi: 10.7551/mitpress/7503.003.0006.

[12] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, Jun. 2018, doi: 10.1109/TCCN.2018.2809722.

[13] Y. Yifei and S. Lakshminarayanan, "Multi-agent reinforcement learning for process control: Exploring the intersection between fields of reinforcement learning, control theory, and game theory," *The Canadian Journal of Chemical Engineering*, vol. 101, no. 11, pp. 6227–6239, Nov. 2023, doi: 10.1002/cjce.24878.

[14] V. Uc-Cetina, N. Navarro-Guerrero, A. Martin-Gonzalez, C. Weber, and S. Wermter, "Survey on reinforcement learning for language processing," *Artificial Intelligence Review*, vol. 56, no. 2, pp. 1543–1575, Feb. 2023, doi: 10.1007/s10462-022-10205-5.

[15] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[16] A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita, "Lexicon-based comments-oriented news sentiment analyzer system," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9166–9180, Aug. 2012, doi: 10.1016/j.eswa.2012.02.057.

[17] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17259–17274, Dec. 2020, doi: 10.1007/s00521-020-04757-2.

[18] F. Janjua, A. Masood, H. Abbas, I. Rashid, and M. M. Z. M. Khan, "Textual analysis of traitor-based dataset through semi supervised machine learning," *Future Generation Computer Systems*, vol. 125, pp. 652–660, Dec. 2021, doi: 10.1016/j.future.2021.06.036.

[19] A. Rajan and M. Manur, "Aspect based sentiment analysis using a novel ensemble deep network," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 2, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1668-1678.

[20] A. Esteso, D. Peidro, J. Mula, and M. Díaz-Madroñero, "Reinforcement learning applied to production planning and control," *International Journal of Production Research*, vol. 61, no. 16, pp. 5772–5789, Aug. 2023, doi: 10.1080/00207543.2022.2104180.

[21] Y.-H. Ho, C.-Y. Cho, and W.-H. Peng, "Deep reinforcement learning for video prediction," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2019, pp. 604–608, doi: 10.1109/ICIP.2019.8803825.

[22] H. Teng, Y. Li, F. Long, M. Xu, and Q. Ling, "Reinforcement learning for extreme multi-label text classification," in *Communications in Computer and Information Science*, 2021, pp. 243–250, doi: 10.1007/978-981-16-2336-3_22.

[23] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations*, 2015, pp. 1-13.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[25] T. V. Bui, T. O. Tran, and P. Le-Hong, "Improving sequence tagging for Vietnamese text using transformer-based neural models," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, Hanoi, Vietnam: Association for Computational Linguistics, Oct. 2020, pp. 13–20.

[26] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 1037–1042, doi: 10.18653/v1/2020.findings-emnlp.92.

BIOGRAPHIES OF AUTHORS

Viet The Bui holds a Master of Software Engineering degree from FPT University, Vietnam. He is currently a Ph.D. student in Computer Science at the School of Computing and Information Systems, Singapore Management University. His research areas include artificial intelligence and data science, with a focus on machine learning and intelligence, reinforcement learning, NLP, and operation research. He has published numerous papers in top-tier international conferences and journals. He has been awarded the President Doctoral Fellowship and the Best Research Staff Award at Singapore Management University. With four years of research experience in deep learning, he continues to contribute significantly to his field. He can be contacted at email: theviet.bui.2023@phdcs.smu.edu.sg.

Linh Thuy Ngo is a Ph.D. student in the Doctor of Philosophy in Informatics and Computer Engineering program. She received a Master's degree in Information Systems from the University of Engineering and Technology, Vietnam National University, Hanoi in 2010 and a Bachelor's degree in IT from the Posts and Telecommunications Institute of Technology in 2004. Her main research areas are artificial intelligence, machine learning, and NLP. She is currently a lecturer at the Faculty of Information Technology and Digital Economics, Banking Academy of Vietnam, Hanoi, Vietnam. She can be contacted at email: linhnt@hvnh.edu.vn or linhnt@bav.edu.vn.

Oanh Thi Tran got her bachelor and master degrees in Computer Science at the University of Engineering and Technology, Vietnam National University, Hanoi in 2006 and 2009, respectively. She was awarded a Japanese Government Scholarship to pursue Ph.D. in Computer Science at Japan Advanced Institute of Science and Technology (JAIST) from 2011 to 2014. Currently, she is a lecturer at the International School of Vietnam National University, Hanoi (VNU-IS). Her main research interests are artificial intelligence and machine learning. Her contributions to the field include 50 publications in esteemed journals and conferences. She can be contacted at email: oanhtt@gmail.com or tranthioanh@vnu.edu.vn.