Neurocomputing 580 (2024) 127423

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

CAT: Continual Adapter Tuning for aspect sentiment classification

Qiangpu Chen [a,1], Jiahua Huang [a,1], Wushao Wen [a], Qingling Li [a], Rumin Zhang [c], Jinghui Qin [b,*]

[a] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, 510006, China
[b] School of Information Engineering, Guangdong University of Technology, Guangzhou, 510006, China
[c] Ningbo Institute Of Digital Twin(EIAS), Ningbo, 31520, China

ARTICLE INFO

Communicated by P. Hájek

Keywords:
Aspect sentiment classification
Continual learning
Adapter tuning

ABSTRACT

Humans can continually acquire, improve, and transfer knowledge throughout their lifespan so that they can accurately identify sentiment polarities of the data attributed to different domains. However, the continual learning of incrementally available aspect sentiment classification (ASC) tasks from different domains with non-stationary data distributions remains a long-standing challenge for learning-based models due to the catastrophic forgetting problem, which is the tendency to completely and abruptly forget previously learned knowledge upon learning new knowledge. In this work, we present Continual Adapter Tuning (CAT), a parameter-efficient framework that not only avoids catastrophic forgetting but also enables knowledge transfer from learned ASC tasks to new ASC tasks. To avoid catastrophic forgetting, we only learn and store a task-specific adapter for each ASC task while freezing the backbone pre-trained model. To promote new task learning, we propose a continual adapter initialization technique to transfer knowledge from preceding tasks. Besides, we also develop a novel label-aware contrastive learning to simultaneously learn the features of input samples and the parameters of classifiers in the same space so that we can efficiently classify a sample with the help of label semantics. To eliminate the need for task IDs in testing, we propose a simple yet efficient majority sentiment polarity voting strategy to obtain final sentiment polarities according to the polarities predicted by all reasoning paths in the adapter architecture. Experimental results show the high effectiveness of our CAT by achieving new state-of-the-art performance.

1. Introduction

Recently, most studies have focused on developing aspect sentiment classification (ASC) models for specific domains by assuming the data distribution stays the same. However, this is far from realistic because input data may belong to a new domain that is not encountered by the current model in practice, which usually makes a deployed ASC model required to support new domains for ensuring the quality of service. Therefore, it is crucial for an ASC model to be able to continually learn new tasks without forgetting old ones with high efficiency so that it can provide high-performance predictions in real scenarios.

In an aspect sentiment classification case, a task is a separate aspect sentiment classification (ASC) problem of a product or domain (e.g., laptop or mobile phone) [1] where ASC is stated as follows: Given an aspect item (e.g., *mobile phone*) and a sentence containing the aspect (e.g., *The mobile phone is good*), ASC aims to classify the sentiment polarity about the aspect implied in the sentence, such as positive, negative or neutral opinion. In the setting of the continual learning for ASC, we hope an ASC model to learn a series of tasks incrementally without forgetting old tasks catastrophically. However, neural networks are prone to suffering from Catastrophic Forgetting (CF) [2,3], which is a key issue that many previous studies [4,5] mainly focused on solving. The catastrophic forgetting problem can be defined as follows: when a neural model is trained on a sequence of tasks, new tasks may interfere catastrophically with old tasks, leading to dramatic performance degradation.

Simply storing a model checkpoint for each task to mitigate catastrophic forgetting is prohibitive as the number of tasks grows, especially when the model is large. To mitigate catastrophic forgetting and storage overhead, recent methods froze the backbone model and proposed to train a weight mask [6,7], a feature mask [8], or an adapter [9] for each task independently. However, they either have limited capacity to support more new tasks or largely ignore knowledge transfer among tasks.

In this paper, unlike the vanilla approach of training each task's adapter from scratch, we propose the parameter-efficient *Continual Adapter Tuning* for continual learning by enabling knowledge transfer

* Corresponding author.
E-mail address: scape1989@gmail.com (J. Qin).
[1] The first two authors contribute equally.

https://doi.org/10.1016/j.neucom.2024.127423
Received 3 May 2023; Received in revised form 22 November 2023; Accepted 17 February 2024
Available online 20 February 2024

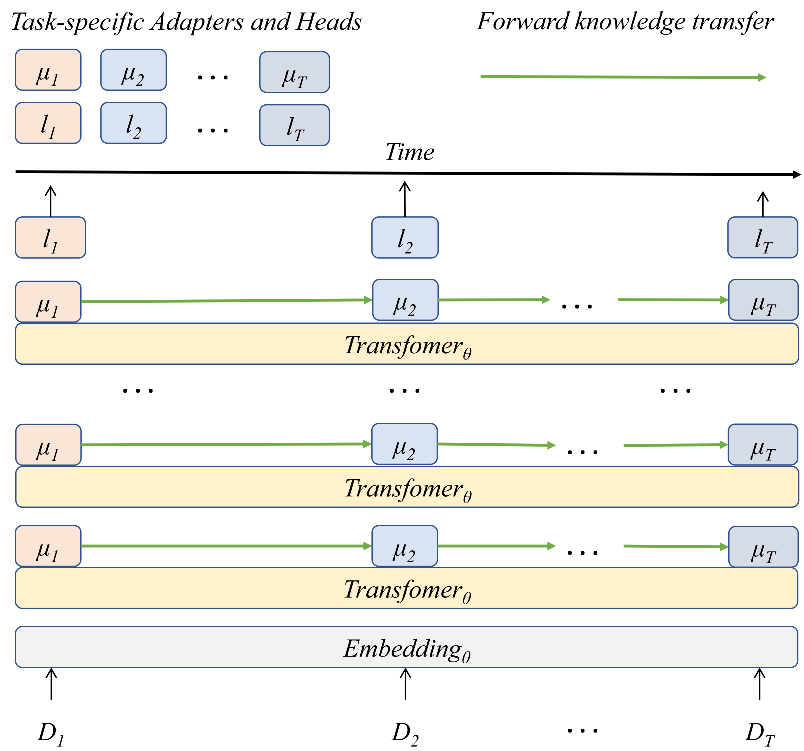


Fig. 1. An illustration of our Continual Adapter Tuning. We train a task-specific adapter (μ_i) and a task-specific classification head (l_i) for each ASC task and freeze the pre-trained model ($Embedding_\theta$ and $Transformer_\theta$). Several adapter initialization techniques are proposed to transfer knowledge from preceding tasks to subsequent tasks (green solid arrows). The task-specific classification head (l_i) is initialized randomly for each ASC task. For notion simplicity, we represent the parameters of an adapter added to different transformer blocks and the parameters of different transformer blocks in the pre-trained model as μ_i and $Transformer_\theta$ uniformly.

between tasks while avoiding catastrophic forgetting, as shown in Fig. 1. Specifically, We freeze the backbone pre-trained model and train an adapter for each ASC task, which is highly parameter-efficient to avoid forgetting. Meanwhile, we consider transferring knowledge from preceding tasks to subsequent tasks so that the subsequent tasks can be promoted by prior knowledge from preceding tasks. To realize forward knowledge transfer, we propose several simple but effective continual adapter initialization techniques, including initializing from the last task, initializing from one of the previous tasks by random selection, and initializing from the best previous task which has minimum validation loss on the current task. Besides, to learn an adapter for each task more efficiently, we also developed a novel label-aware contrastive learning to simultaneously learn the features of input samples and the parameters of classifiers in the same space so that we can identify sentiment polarities with the help of label semantics more efficiently. Since the task ID is agnostic at testing, to eliminate the need for task IDs in testing, we propose a simple yet efficient majority sentiment polarity voting strategy to obtain final sentiment polarity according to the polarities predicted by all reasoning paths in the adapter architecture. Experimental results on 19 ASC task datasets show the effectiveness of our CAT by achieving a new state-of-the-art performance.

Overall, the main contributions of our work can be summarized as follows:

- We propose the parameter-efficient *Continual Adapter Tuning* for continual learning by enabling knowledge transfer between tasks while avoiding catastrophic forgetting. Continual Adapter Tuning is implemented through a frozen pre-trained model, task-specific adapters, and forward knowledge transfer.
- To realize forward knowledge transfer, we propose several simple but effective continual adapter initialization techniques, including initializing from the last task, from one of the previous tasks by random selection, and from the best previous task which has minimum validation loss on the current task.
- To learn an adapter for each task more efficiently, we develop a novel label-aware contrastive learning to simultaneously learn the features of input samples and the parameters of classifiers in the same space so that we can identify sentiment polarities with the help of label semantics more efficiently.
- To eliminate the need for task IDs in testing, we propose a simple yet efficient majority sentiment polarity voting strategy to obtain final sentiment polarity according to the polarities predicted by all reasoning paths in the adapter architecture.
- Experimental results on 19 ASC task datasets show the effectiveness of our CAT by achieving a new state-of-the-art performance.

2. Related work

Aspect Sentiment Classification. Aspect sentiment classification is typically regarded as a text classification problem. Therefore, text classification approaches [10–12] can be naturally applied to solve the aspect sentiment classification task. Besides, deep learning approaches have shown promising results on sentiment classification in recent years, such as Recursive NN [13], Recursive NTN [14] and Tree-LSTM [15]. However, these deep learning-based methods only make use of the sentence contexts without consideration of aspects that make great contributions to identifying the sentiment polarity.

Therefore, to incorporate aspects into a model, Tang et al. [16] proposed two LSTM to model the left and right contexts with the target. Wang et al. [17] proposed an attention-based LSTM to explore the potential correlation of aspects and sentiment polarities. Chen et al. [18] designed deep memory networks to integrate the target information. Ma et al. [19] proposed an interactive learning approach to interactively learn attention between the contexts and targets. Wang et al. [20] addressed aspect sentiment classification with both word-level attention and clause-level attention. Chen et al. [21] proposed

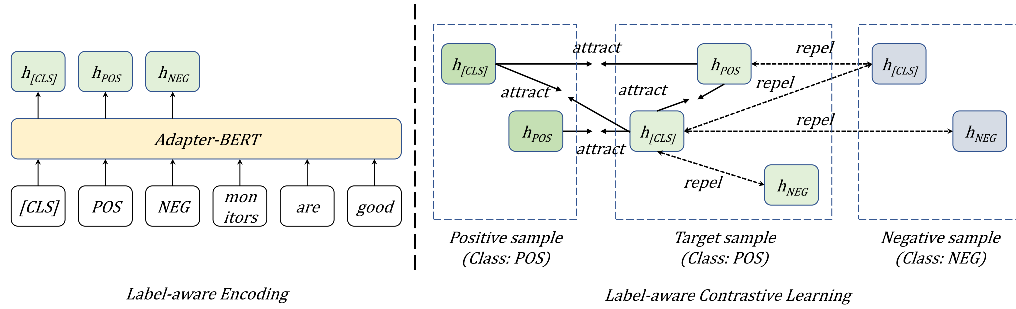


Fig. 2. Label-aware aspect sentiment classification. Left: Label-aware Encoding. We inject the list of sentiment labels after the [CLS] token. Right: Label-aware Contrastive Learning. We pull the input features and their corresponding label features closer and push away the input features and other label features.

a Transfer Capsule Network (TransCap) model to transfer document-level knowledge to aspect-level sentiment classification. Zeng et al. [22] proposed a novel relation construction multi-task learning network (RMN) to make the first attempt to extract aspect relations as an auxiliary classification task for aspect sentiment classification.

Although the above deep neural network models have achieved great success on aspect sentiment classification, they all ignore the realistic requirement to support new domains in real scenarios. In this paper, orthogonal to the above methods, we focus on the domain incremental learning for aspect sentiment classification with an improved Adapter-BERT. Besides, to learn an adapter for each task more efficiently by injecting label semantics, we also develop a novel label-aware contrastive learning inspired by DualCL [12] to learn the features of input samples and the parameters of classifiers in the same space simultaneously.

Continual Learning. Continual, or lifelong, learning aims to learn from a sequence of tasks incrementally [23]. However, when re-trained deep networks with new tasks, they suffer from a challenge termed catastrophic forgetting [2,3], which tends to forget how to perform previous tasks. Several techniques have been proposed to mitigate forgetting [4,5,24–27]. However, their memory is imperfect. LwF [5] avoided catastrophic forgetting by training both old tasks and new tasks with the data of new tasks. At testing, it also required each sample to be accompanied by the information of the task it belongs to. Progressive Networks [28] avoided catastrophic forgetting by instantiating a new network path for each task, but the number of parameters grows linearly with the number of tasks. To parameter-efficient continual learning, Adapter-BERT [29] inserted a 2-layer fully connected network (adapter) in each transformer layer of BERT. During training for the end task, only the adapters and normalization layers were trained, with no change to any other BERT parameters, which is good for CL. However, it ignores the knowledge transfer from previous tasks to new tasks, which can help new adapters converge and generalize better. Besides, it also requires task IDs for testing. Although both AFPKT [30] and CLASSIC [7] avoid catastrophic forgetting while enabling knowledge transfer across tasks by weight masking, they have limited capacity to support more new tasks.

In this work, we first adopt Adapter-BERT as the basic architecture to avoid catastrophic forgetting and develop several simple but effective continual adapter initialization techniques to achieve knowledge transfer. Then, we develop a novel label-aware contrastive learning to simultaneously learn the features of input samples and the parameters of classifiers in the same space so that we can identify sentiment polarities with the help of label semantics more efficiently. Finally, we eliminate the need for task IDs in testing by proposing a simple yet efficient majority sentiment polarity voting strategy. It can obtain final sentiment polarity according to the polarities predicted by all reasoning paths in the adapter architecture.

3. Continual Adapter Tuning (CAT)

3.1. Overview of CAT

The goal of continual learning is to sequentially learn a model $f : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{Y}$ from a stream of tasks $\mathcal{T}_1 \dots \mathcal{T}_T$ that can predict the target y given the input x and task $\mathcal{T}_i \in \mathcal{T}$. The data for each task $\mathcal{T}_i$ is denoted as $\mathcal{D}_i$. The overview architecture of CAT is given in Fig. 1, which works in the domain incremental learning setting for ASC. It uses Adapter-BERT [29] to freeze BERT and learns one adapter for each task so that it can avoid catastrophic forgetting. Our CAT takes a sentence and an aspect item as input and outputs a hidden state $\boldsymbol{h}_{[CLS]}$ and label-aware features $\left[l_1, \dots, l_K\right]$ for task $\mathcal{T}_i$ to build a classifier where K is the number of classes of task $\mathcal{T}_i$. How to build a classifier with the hidden state $\boldsymbol{h}_{[CLS]}$ and label-aware features $\boldsymbol{l} = \left[l_1, \dots, l_K\right]$ for task $\mathcal{T}_i$ will be introduced in the next section.

3.2. Label-aware ASC

Many existing works [12,31] show that the semantics of labels are important for a classifier and should be considered in the same space of input feature to learn a more efficient classifier. Inspired by them, we develop a label-aware classifier for each task with contrastive learning to simultaneously learn the features of input samples and the parameters of adapters and classification heads in the same space so that we can identify sentiment polarities more efficiently with the help of label semantics.

3.2.1. Label-aware classification head

To construct a label-aware classification head for our CAT, we first concatenate the input sentence (e.g., *"The price is very good"*.) and the corresponding aspect term (e.g., *price*) with [SEP] as input and insert [CLS] token at the beginning of the input. Then the list of sentiment labels is inserted after the [CLS] token (e.g., *"positive negative"* for binary classification or *"positive negative neutral"* for 3-classification), as shown in the left part of Fig. 2.

Let $\boldsymbol{h}_i \in \mathbb{R}^d$ be the representation feature of an input example x_i, which is obtained from the feature of the [CLS] token and $\boldsymbol{l}_i \in \mathbb{R}^{d \times K}$ be the classifier associating to x_i where d is the feature dimension of Adapter-BERT and K is the number of classes. The logits of sentiment polarities can be model as $\boldsymbol{l}_i^T \boldsymbol{h}_i$ where $\boldsymbol{l}_i^T$ is the transposed version of $\boldsymbol{l}_i$. Thus, the sentiment polarity prediction $\hat{y}_i$ for x_i can be computed as follow:

$$\hat{y}_i = \arg\max_k \left(\boldsymbol{l}_i^k \cdot \boldsymbol{h}_i\right) \tag{1}$$

where $\boldsymbol{l}_i^k$ is the classifier parameters of kth sentiment polarity label.

3.2.2. Label-aware contrastive learning

Contrastive learning [32] insists the feature representations of samples in the same class to be similar and those for different classes to be distinct. It has been shown to improve task performance by making more discriminative feature representations with feature alignment [33]. In our task, we deploy a label-aware classification head as the sentiment polarity classifier. To build a more discriminative feature representation, it is important to align not only the input feature and the classifier associated with an input sample but also the input feature of a sample and the input features of the input sample's positive samples. Therefore, we propose label-aware contrastive learning to achieve this goal. An intuitive example is shown in the right part of Fig. 2.

To fully exploit the semantic relationships between the input feature $\boldsymbol{h}_i$ and the classifier $\boldsymbol{l}_i$ associating to x_i, we align the softmax transform of $\boldsymbol{l}_i^T \boldsymbol{h}_i$ with the sentiment polarity label of x_i. Let $\boldsymbol{l}_i^*$ denote the column of $\boldsymbol{l}_i$, corresponding to the ground-truth label of x_i. Since we aim to align $\boldsymbol{h}_i$ and $\boldsymbol{l}_i^*$, we try to maximize the dot product $(\boldsymbol{l}_i^*)^T \cdot \boldsymbol{h}_i$ so that we can learn a better representation of $\boldsymbol{h}_i$ and $\boldsymbol{l}_i$. To avoid interference from other different labels with x_i, we try to minimize $(\boldsymbol{l}_i^k)^T \cdot \boldsymbol{h}_i$ where $k \neq *$. Similarly, to exploit the relation between different training samples, we try to maximize $(\boldsymbol{l}_i^*)^T \boldsymbol{h}_j$ and $\boldsymbol{h}_i \boldsymbol{h}_j$ if another sample x_j has the same sentiment polarity label with x_i while minimizing $\boldsymbol{l}_i^{*T} \boldsymbol{h}_j$ and $\boldsymbol{h}_i \boldsymbol{h}_j$ if x_j has a different label with x_i.

We assume there are N training samples $\{x_i\}_{i=1}^N$ and denote the set of indexes of the training sample by $\mathcal{I} = \{1, 2, \dots, N\}$. Given an anchor $\boldsymbol{h}_i$ originated from sample x_i, from classifie's view, we take $\left\{\boldsymbol{l}_p^*\right\}_{p \in \mathcal{P}_i}$ as positive samples and $\left\{\boldsymbol{l}_j^*\right\}_{j \in \mathcal{A}_i}$ as negative samples where $\mathcal{P}_i$ is the set of indexes of positive samples and $\mathcal{A}_i$ be the set of indexes of negative samples. From the feature's view, we take $\mathcal{P}_i$ to be the set of indexes of the samples that have the same labels with x_i, $\mathcal{A}_i$ to be the set of indexes of negative samples which have different labels with x_i. Then we can define the following contrastive loss for the feature-level anchor:

$$
\begin{aligned}
\mathcal{L}_{\boldsymbol{h}} = \frac{1}{N} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} \Big(& - log \frac{exp((\boldsymbol{h}_i \cdot \boldsymbol{l}_p^*)/\tau)}{\sum_{j \in \mathcal{A}_i} exp((\boldsymbol{h}_i \cdot \boldsymbol{l}_j^*)/\tau)} \\
& - log \frac{exp((\boldsymbol{h}_i \cdot \boldsymbol{h}_p)/\tau)}{\sum_{j \in \mathcal{A}_i} exp((\boldsymbol{h}_i \cdot \boldsymbol{h}_j)/\tau)} \Big)
\end{aligned} \tag{2}
$$

Similarly, given an anchor $\boldsymbol{l}_i^*$, we can also take $\left\{\boldsymbol{h}_p\right\}_{p \in \mathcal{P}_i}$ as positive samples and $\left\{\boldsymbol{h}_j\right\}_{j \in \mathcal{A}_i}$ as negative samples. Then, we can define the contrastive loss for the classifier-level anchor as follows:

$$
\mathcal{L}_{\boldsymbol{l}} = \frac{1}{N} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} -log \frac{exp((\boldsymbol{h}_p \cdot \boldsymbol{l}_i^*)/\tau)}{\sum_{j \in \mathcal{A}_i} exp((\boldsymbol{h}_j \cdot \boldsymbol{l}_i^*)/\tau)} \tag{3}
$$

Thus, the overall label-aware contrastive loss is a combination of $\mathcal{L}_{\boldsymbol{h}}$ and $\mathcal{L}_{\boldsymbol{l}}$:

$$
\mathcal{L}_{CL} = \mathcal{L}_h + \mathcal{L}_{\boldsymbol{l}} \tag{4}
$$

3.2.3. Training

Let $\boldsymbol{l}_i$ be a good classifier for $\boldsymbol{h}_i$. To fully exploit the supervised signal, we train Adapter-BERT with a variant cross-entropy loss to maximize the dot product $\boldsymbol{l}_i^{*T} \boldsymbol{h}_i$ for each input sample x_i. The sentiment polarity prediction $\hat{y}_i$ for x_i is decided by the dot product $\boldsymbol{l}_i^{*T} \boldsymbol{h}_i$. Therefore, the variant cross-entropy loss can be defined as follows:

$$
\mathcal{L}_{CE} = \frac{1}{N} \sum_{i \in \mathcal{I}} -log \frac{exp((\boldsymbol{l}_i^* \cdot \boldsymbol{h}_i))}{\sum_{k \in \mathcal{K}} exp((\boldsymbol{l}_i^k \cdot \boldsymbol{h}_i))} \tag{5}
$$

Finally, since variant cross-entropy loss and the label-aware contrastive loss simultaneously improve the quality of the representations of the features and the classifiers, we minimize them to train the Adapter-BERT for each ASC task as follows:

$$
\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{CL} \tag{6}
$$

where λ is a hyperparameter that adjusts the influence of the label-aware contrastive loss. We set 0.1 as the default value of the λ empirically.

3.3. Continual adapter initialization

An intuitive and simple way to transfer knowledge is the adapter's parameter initialization. Therefore, we explore three empirically effective continual adapter initialization strategies as shown in Fig. 3:

- **LastInit** uses the last task's adapter μ_{t-1} to initialize the current task's adapter μ_t. In LastInit, the latest adapter has been continually trained on all previous tasks.
- **RandomInit** randomly chooses one of the trained tasks μ_i $(i < t)$ to initialize the current task's adapter μ_t. The RandomInit only uses a random trained adapter to train a new task.
- **SelectInit** evaluates all $\{\mu_i\}_{i<t}$ on the validation set of current task $\mathcal{T}_t$ without training and selects the one adapter with the lowest loss to initialize μ_t. The SelectInit considers the most relevant task without interference from its subsequent tasks.

We will empirically compare these three strategies in the Experiment section.

3.4. Sentiment polarity voting

The Adapter-BERT needs the task ID to choose the appropriate adapter for test data. However, the task ID for the data is agnostic at the test. Therefore, to eliminate the need for task IDs in testing, we propose a simple yet efficient majority sentiment polarity voting strategy to obtain final sentiment polarity according to the polarities predicted by all reasoning paths in the adapter architecture.

Assuming there are T well-trained adapters which means that there are T reasoning paths, we forward a test data x_i into all T reasoning paths. Then we can obtain T predictions about x_i's sentiment polarity. Assuming the number of sentiment classes is 2 (e.g., Positive and Negative), we count the number of positive polarities and the number of negative polarities in all prediction results. We use the polarity with maximum count as the final prediction for x_i. If the number of positive polarities is more than the number of negative polarities, we predict x_i as positive. Otherwise, we predict x_i as negative. If the number of positive polarities is equal to the number of negative polarities, we use the prediction with maximum confidence as the final prediction. The voting algorithm is shown in Algorithm 1.

4. Experiments

We evaluate the proposed CAT framework in this section. We first introduce the experiment settings about datasets, metrics, baselines, and implement details. Then, we compare CAT with both non-continual learning and continual learning baselines. Finally, we conduct some ablation experiments to show the impact of different modules in our CAT.

4.1. Experiments datasets

For a fair comparison, we also use 19 ASC datasets introduced in [7] to produce sequences of 19 tasks. Each dataset is a set of aspects and annotated comment sentences from comments of a particular product and represents a task. These datasets are from 4 different sources: (1) SemEval14 [34]: comment sentences of laptop and restaurant; (2) Liu3Domains [35]: comment sentences of 3 products; (3) HL5Domains [36]: comment sentences from 5 products; (4) Ding9Domains [37]: comment sentences from 9 products. The detailed statistics of the 19 ASC datasets are shown in Table 1.

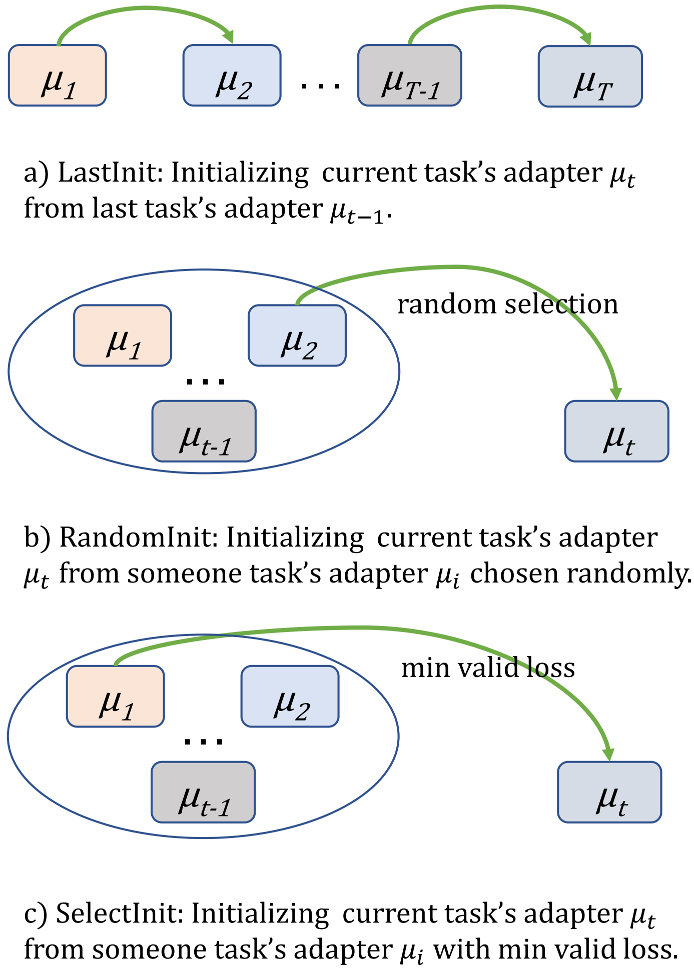


Fig. 3. Three continual adapter initialization techniques for knowledge transfer.

Table 1
Number of samples in each task or dataset.

Source	Task/Domain	Train	Validation	Test
SemEval14	Rest.	3452	150	1120
	Laptop	2163	150	638
Liu3Domain	Nokia	352	44	44
	Router	245	31	31
	Computer	283	35	36
HL5Domain	Nokia6610	271	34	34
	Nikon4300	162	20	21
	CreativeNomad	677	85	85
	CanonG3	228	29	29
	ApexAD	343	43	43
Ding9Domain	CanonD500	118	15	15
	Canon100	175	22	22
	Diaper	191	24	24
	Hitachi	212	26	27
	Ipod	153	19	20
	Linksys	176	22	23
	MicroMP3	484	61	61
	Nokia6600	362	45	46
	Norton	194	24	25

4.2. Evaluation protocol and metrics

For evaluation metrics, we compute both accuracy and Marco-F1. Macro-F1 is a better metric than accuracy since the imbalanced classes introduce biases in accuracy.

For evaluation protocol, we follow the standard CL evaluation method in previous works [7,38,39]. We first present a sequence of ASC tasks for CAT to learn. Once a task is learned, we discard its training data and evaluate its test set by the current learned model with accuracy and Marco-F1. We call this accuracy and Marco-F1 as forward accuracy and forward Macro-F1. We evaluate all the previous tasks' test sets with our Sentiment Polarity Voting mechanism to obtain backward accuracy and backward Marco-F1 which these two metrics can show the effect of our CAT in real scenarios. Similarly, after all tasks are learned, we evaluate our CAT using the test data of all tasks without giving any task ID.

As the order of the 19 ASC tasks can influence, we randomly select and run 5 task sequences and report their average results and standard deviations.

4.3. Baselines

We employ 26 baselines including both non-continual learning and continual learning approaches. These baselines are adapted from task-incremental learning (TIL) models to domain-incremental learning (DIL) settings by sharing one classification head. The brief introduction of these baselines is as follows:

Non-Continual Learning Baselines: We employ 3 non-continual learning baselines: (1) **BERT** (Frozen) without fine-tuning, (2) **BERT** which is not frozen, (3) **Adapter-BERT** [29], (4) **BERT** (Multi) which is trained in multi-task mode. Each of these baselines builds a separate

Algorithm 1 Sentiment Polarity Voting

Input: input data x_i of T-th task's test set, all T well-trained adapters $M_T = \{\mu_1 .., \mu_T\}$.
Output: final polarity prediction of input data x_i

for $k = 1, 2, ..., T$ **do**
  Load adapter μ_k into Adapter-BERT θ
  $\hat{y}_k = \theta(x_i)$
  $p_k = argmax(SoftMax(\hat{y}_k))$
end for
let $pos\_cnt, neg\_cnt = 0{,}0$
for $k = 1, 2, ..., T$ **do**
  pos_cnt += 1 if $p_k == pos$
  neg_cnt += 1 if $p_k == neg$
end for
if pos_cnt != neg_cnt **then**
  $cnt = array([pos\_cnt, neg\_cnt])$
  $P_i = max(cnt)$
else
  $pos\_dif = \sum_{p \in \{k|p_k == pos\}} |y_{k,pos} - y_{k,neg}|$
  $neg\_dif = \sum_{p \in \{k|p_k == neg\}} |y_{k,neg} - y_{k,pos}|$
  $dif = array([pos\_dif, neg\_dif])$
  $P_i = max(dif)$
end if
return P_i

model for each task independently without sharing any knowledge and catastrophic forgetting (CF). We tag these baselines as ONE variant.

Continual Learning Baselines: In a continual learning setting, there are 23 baselines in 6 categories. The first category uses a naive CL (NCL) approach which simply deploys a network to continually learn all tasks with no mechanism to overcome CF or achieve knowledge transfer. In this category, like ONE, there also are 3 NCL variants. The second category has 9 BERT(Frozen)-based baselines with recent proposed CL methods: KAN [40], SRK [41], HAT [42], UCL [43], EWC [4], OWM [44], and DER++ [45]. In these methods, KAN and SRK are for document sentiment classification. The input is the concatenation of the aspect and the sentence. HAT, UCL, EWC, OWM, and DER++ are designed for image classification. To adapt to the ASC task, their original image classification networks are replaced with CNN for text classification [46]. There are 2 variants of HAT and KAN: (1) *+last* means using the *last* model in testing, (2) *ent* means detecting task ID using the entropy method in [47]. The third category has 6 baselines using Adapter-BERT as the base. The LAMOL uses the GPT-2 model as the base. The B-CL [9] and CLASSIC [7] are the most recent methods dealing with ASC tasks.

4.4. Implementation details

We adopt $\text{BERT}_{\text{base}}$ (uncased) as our backbone pre-trained model. The adapter uses 2 layers of a fully connected network with dimensions 256, followed by the ReLU activation function. The dropout probability is set to 0.1 and the temperature τ is set to 0.1. We use the AdamW optimizer with a weight decay of 0.01 and set the initial learning rate to 2e−5. For SemEval datasets, we train the adapters 20 epochs, and for all other datasets, the adapters are trained 30 epochs. The batch size for all tasks is set to 32.

4.5. Results and analysis

The main results are shown in Tables 2 and 3. From Tables 2 and 3, we can observe that all our CAT with different transfer strategies outperform all baselines.

Table 2
Accuracy and Macro-F1 averaged over 5 random sequences of 19 tasks compared to the non-continual learning baselines.

Category	Model	Acc.	Macro-F1
BERT(Frozen)	ONE	0.7814	0.5813
BERT(No. Frozen)	ONE	0.8584	0.7635
Adapter-BERT	ONE	0.8530	0.7516
BERT(Multi)	ONE	0.9066	0.8596
CAT-LastInit (forward)		0.9030	0.8536
CAT-LastInit (backward)		**0.9107**	**0.8673**
CAT-RandomInit (forward)		0.9005	0.8457
CAT-RandomInit (backward)		**0.9056**	**0.8607**
CAT-SelectInit (forward)		0.9040	0.8539
CAT-SelectInit (backward)		**0.9110**	**0.8722**

Table 3
Accuracy and Macro-F1 averaged over 5 random sequences of 19 tasks compared to the continual learning baselines.

Category	Model	Acc.	Macro-F1
	KAN+last	0.8320	0.7352
	KAN+ent	0.8278	0.7243
	SRK	0.8391	0.7438
	EWC	0.8660	0.7831
BERT (Frozen)	UCL	0.8538	0.7690
	OWM	0.8611	0.7665
	DER++	0.8753	0.8009
	HAT+last	0.8473	0.7649
	HAT+ent	0.8418	0.7614
	EWC	0.8805	0.7875
	UCL	0.7123	0.3961
	OWM	0.8766	0.7882
Adapter-BERT	DER++	0.8859	0.7985
	HAT+last	0.8823	0.7919
	HAT+ent	0.8854	0.8245
LAMOL		0.8891	0.8059
B-CL (forward)		0.8809	0.7993
B-CL (backward)		0.8829	0.8140
CLASSIC (forward)		0.8886	0.8365
CLASSIC (backward)		0.9022	0.8512
CAT-LastInit (forward)		0.9030	0.8536
CAT-LastInit (backward)		**0.9107**	**0.8673**
CAT-RandomInit (forward)		0.9005	0.8457
CAT-RandomInit (backward)		**0.9056**	**0.8607**
CAT-SelectInit (forward)		0.9040	0.8539
CAT-SelectInit (backward)		**0.9110**	**0.8722**

On forward evaluation, our **CAT-LastInit (forward)**, **CAT-RandomInit (forward)**, and **CAT-SelectInit (forward)** outperform current state-of-the-art model CLASSIC (forward) up to 1.44%, 1.19%, and 1.54% in accuracy, respectively. Similarly, our **CAT-LastInit (forward)**, **CAT-RandomInit (forward)**, and **CAT-SelectInit (forward)** outperform CLASSIC (forward) up to 1.71%, 0.92%, and 1.74% in Marco-F1, respectively. These results show that our three different continual adapter initialization strategies (i.e., **LastInit**, **RandomInit**, and **SelectInit**) are effective for transferring learned knowledge into new tasks. Besides, they show the effectiveness of our label-aware ASC with label-aware contrastive learning.

On backward evaluation which is consistent with the realistic requirement of continual learning, we can observe that our **CAT-LastInit (backward)**, **CAT-RandomInit (backward)**, and **CAT-SelectInit (backward)** outperforms current state-of-the-art model CLASSIC (backward) over 0.85%, 0.34%, and 0.88% on the accuracy, respectively. Similarly, our **CAT-LastInit(backward)**, **CAT-RandomInit (backward)**, and **CAT-SelectInit (backward)** outperform CLASSIC(backward) up to 1.58%, 0.95%, and 2.09% on Marco-F1, respectively. These results show that our CAT can overcome catastrophic forgetting better. Besides, these results also demonstrate the effectiveness of our sentiment polarity voting mechanism which eliminates the need for task IDs at testing in the DIL setting.

Table 4
Ablation experiment results of different strategies for continual adapter initialization.

Strategy	Model	Acc.	Macro-F1
LastInit	**CAT**	**0.9107**	**0.8673**
	-CL	0.9087	0.8663
	-SPV	0.9022	0.8439
	-CL-SPV	0.9011	0.8469
RandomInit	**CAT**	**0.9056**	**0.8607**
	-CL	0.9053	0.8594
	-SPV	0.8892	0.8360
	-CL-SPV	0.8909	0.8382
SelectInit	**CAT**	**0.9110**	**0.8722**
	-CL	0.9049	0.8600
	-SPV	0.8911	0.8401
	-CL-SPV	0.8974	0.8482

Table 5
Results of ablation experiment of prediction head, the label-aware is based on SelectInit strategy.

Prediction Head	Model	Acc.	Macro-F1
Label-Aware classification head	**CAT**	**0.9110**	**0.8722**
	-CL	0.9087	0.8663
	-SPV	0.9022	0.8439
	-CL-SPV	0.9011	0.8469
Linear classification head	CAT	–	–
	-CL	**0.8991**	**0.8421**
	-CL-SPV	0.8882	0.8292

Table 6
Results of ablation experiment of different λ in the loss $\mathcal{L}$.

λ	Acc.	Macro-F1
0.0	0.9049	0.8600
0.01	0.9078	0.8631
0.05	0.9092	0.8668
0.1	**0.9110**	**0.8722**
0.5	0.9073	0.8624
1.0	0.9069	0.8552

Overall, compared to baselines, our CAT can achieve better performance on continual domain-incremental learning of aspect sentiment classification.

4.6. Ablation studies

4.6.1. Effects of different transfer strategies

We investigate the effects of different forward knowledge transfer strategies for CAT. The experimental results are shown in Table 3. We can observe that all three forward knowledge transfer strategies are effective so that our CAT equipped with any forward knowledge transfer strategy can achieve new state-of-the-art performance on the continual learning of aspect sentiment classification.

4.6.2. Effects of different modules

To investigate the effects of different modules, we use "-CL", "-SPV", and "-CL-SPV" to represent the settings removing label-aware contrastive learning, removing sentiment polarity voting mechanism in backward evaluation, and removing both label-aware contrastive learning and sentiment polarity voting mechanism, respectively. We conduct experiments on three different CAT models with different knowledge transfer strategies. The experiment results are shown in Table 4. From the results of Table 4, we can observe that each component is effective and the full CAT models can always achieve the best performance.

4.6.3. Effects of different prediction ways

Different from those classification models that apply a classification head such as a linear layer or MLP to identify the sentiment polarity, our CAT predicts the sentiment polarity based on softmax transform of dot product output $\boldsymbol{l}_i^T \boldsymbol{h}_i$ between the feature representation produced from [CLS] and the label-aware semantic feature produced from label set. To show the effectiveness of our label-aware ASC, we modify the prediction head in CAT to be the linear classification head based on the feature representation produced from [CLS] and compare our CAT with it. The experimental results are shown in Table 5. In the way of linear classification heads, we run the CAT without label-aware contrastive learning since label-aware contrastive learning is not suited for linear classification heads. We also run a model without both label-aware contrastive learning and sentiment polarity voting mechanisms. For the label-aware classification head, we conduct experiments on CAT-SelectInit. From the results, we can observe that the label-aware ASC is more effective than the way with linear classification head.

4.7. Effects of different λ values in the loss $\mathcal{L}$

To investigate the effects of different λ values in the loss function $\mathcal{L}$, we investigate the effects of the CAT-SelectInit models with 6 different λ values, including 0, 0.01, 0.05, 0.1, 0.5, and 1.0. The experimental results about backward accuracy and backward Marco-F1 are in Table 6. We can observe that all our CAT-SelectInit models can achieve high performance on the continual learning of aspect sentiment classification. When λ is set to 0.1, the CAT-SelectInit model achieves the best performance among our studied models with various λ values. Therefore, we set the λ as 0.1 in default.

4.8. Execution time and number of parameters

To show that our method is parameter-efficient, we compare its number of parameters and its average training execution time per task with baselines. Table 7 reports the overall number of parameters (both trainable and non-trainable) and training execution time of different models. The execution time is computed as the average training time per task. Our experiments were run on GeForce RTX 2080 Ti with 11G GPU memory. We trained 20 epochs for SemEval datasets and 30 epochs for other datasets. The batch size was set to 32. From Table 7, we can observe that: First, our method has fewer parameters than Adapter-BERT because our adapter uses 2 layers of a fully connected network with dimensions 256 while Adapter-BERT uses 2 layers of a fully connected network with dimensions 2000 [7]. Therefore, our CAT is more parameter-efficient than Adapter-BERT based on the parameter size, accuracy, and Macro-F1 reported in Tables 2 and 3. Second, compared to the state-of-the-art CLASSIC [7], our CAT has better performance with fewer parameter sizes and less run time according to the results in Tables 3 and 7. Third, compared to other baselines, our CAT achieves a better trade-off among classification performance, parameter sizes, and training cost according to the results reported in Tables 2, 3, and 7. Overall, our CAT is more parameter-efficient.

5. Limitations

In this work, we propose a simple yet effective parameter-efficient framework for continual aspect sentiment classification under a domain-incremental learning setting and achieve new state-of-the-art performance on accuracy and Marco-F1. However, the design of our CAT is focused on domain-incremental learning settings and we still need to explore the effectiveness of our CAT in other continual learning settings. We will explore it in our future work.

6. Conclusion

In this work, we present a parameter-efficient framework named Continual Adapter Tuning (CAT) that not only avoids catastrophic

Table 7
Network parameter's size (trainable and non-trainable) and average run time for training per task of each model.

Scenario	Category	Model	#parameters (M)	run time (s)
Non-CL	BERT (Frozen)	ONE	110.4	87.2
	BERT (No. Frozen)	ONE	109.5	252.7
	Adapter-BERT	ONE	183.3	306.0
	BERT (Multi)	MTL	112.6	134.7
CL	BERT (Frozen)	NCL	110.4	88.3
	BERT (No. Frozen)	NCL	109.5	253.5
	Adapter-BERT	NCL	183.3	307.5
	BERT (Frozen)	KAN	116.6	161.2
		SRK	117.8	1236.4
		EWC	110.4	163.7
		UCL	110.4	276.5
		OWM	110.6	274.9
		DER++	115.0	618.2
		HAT	111.3	92.6
	Adapter-BERT	EWC	183.3	610.2
		UCL	183.4	539.4
		OWM	184.4	481.6
		DER++	184.0	830.0
		HAT	185.2	427.1
	LAMOL		124.4	686.0
	CLASSIC		185.2	949.7
	CAT-LastInit (Ours)		139.0	109.4
	CAT-RandomInit (Ours)		139.0	109.2
	CAT-SelectInit (Ours)		139.0	114.8

forgetting but also enables knowledge transfer from learned ASC tasks to new ASC tasks. To avoid catastrophic forgetting, we only learn and store a task-specific adapter for each ASC task while freezing the backbone pre-trained model. To promote new task learning, we propose a continual adapter initialization technique to transfer knowledge from preceding tasks. Besides, we develop a novel label-aware contrastive learning to simultaneously learn the features of input samples and the parameters of classifiers in the same space so that we can efficiently classify a sample with the help of label semantics. To eliminate the need for task IDs in testing, we propose a simple yet efficient majority sentiment polarity voting strategy to obtain final sentiment polarities according to the polarities predicted by all reasoning paths. Experimental results show the effectiveness of our CAT by achieving state-of-the-art performance.

CRediT authorship contribution statement

Qiangpu Chen: Conceptualization, Investigation, Validation, Writing – original draft. **Jiahua Huang:** Conceptualization, Data curation, Investigation, Methodology, Software, Validation. **Wushao Wen:** Funding acquisition, Resources, Supervision, Writing – review & editing. **Qingling Li:** Data curation, Investigation. **Rumin Zhang:** Data curation, Investigation. **Jinghui Qin:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code and data will be released in github.

Acknowledgements

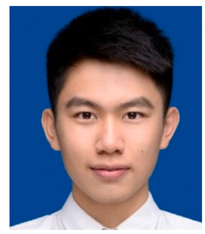
This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62206314 and Grant No. U1711264, GuangDong Basic and Applied Basic Research Foundation under Grant No. 2022A1515011835, China Postdoctoral Science Foundation funded project under Grant No. 2021M703687.

References

[1] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167.
[2] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, vol. 24, Elsevier, 1989, pp. 109–165.
[3] R.M. French, Catastrophic forgetting in connectionist networks, Trends Cogn. Sci. 3 (4) (1999) 128–135.
[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.
[5] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947.
[6] A. Mallya, D. Davis, S. Lazebnik, Piggyback: Adapting a single network to multiple tasks by learning to mask weights, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 67–82.
[7] Z. Ke, B. Liu, H. Xu, L. Shu, CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 6871–6883.
[8] B. Geng, F. Yuan, Q. Xu, Y. Shen, R. Xu, M. Yang, Continual learning for task-oriented dialogue system with iterative network pruning, expanding and masking, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2021, pp. 517–523.
[9] Z. Ke, H. Xu, B. Liu, Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 4746–4755.
[10] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification? in: China National Conference on Chinese Computational Linguistics, Springer, 2019, pp. 194–206.
[11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning–based text classification: a comprehensive review, ACM Comput. Surv. 54 (3) (2021) 1–40.
[12] Q. Chen, R. Zhang, Y. Zheng, Y. Mao, Dual contrastive learning: Text classification via label-aware data augmentation, 2022, arXiv preprint arXiv:2201.08702.
[13] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, C.D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 151–161.
[14] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[15] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1556–1566.
[16] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.
[17] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.
[18] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.
[19] D. Ma, S. Li, X. Zhang, H. Wang, Interactive Attention Networks for Aspect-Level Sentiment Classification.
[20] J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, G. Zhou, Aspect sentiment classification with both word-level and clause-level attention networks, in: IJCAI, vol. 2018, 2018, pp. 4439–4445.
[21] Z. Chen, T. Qian, Transfer capsule network for aspect level sentiment classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 547–556.
[22] J. Zeng, T. Liu, W. Jia, J. Zhou, Relation construction for aspect-level sentiment classification, Inform. Sci. 586 (2022) 209–223.

[23] S. Thrun, L. Pratt, Learning to Learn, Springer Science & Business Media, 2012.
[24] A. Rannen, R. Aljundi, M.B. Blaschko, T. Tuytelaars, Encoder based lifelong learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1320–1328.
[25] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: Learning what (not) to forget, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018.
[26] D. Lopez-Paz, M.A. Ranzato, Gradient episodic memory for continual learning, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.
[27] C. de Masson d'Autume, S. Ruder, L. Kong, D. Yogatama, Episodic memory in lifelong language learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.
[28] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, 2016, arXiv preprint arXiv:1606.04671.
[29] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.
[30] Z. Ke, B. Liu, N. Ma, H. Xu, L. Shu, Achieving forgetting prevention and knowledge transfer in continual learning, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 22443–22456.
[31] T. Miyazaki, K. Makino, Y. Takei, H. Okamoto, J. Goto, Label embedding using hierarchical structure of labels for twitter classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, (EMNLP-IJCNLP), 2019, pp. 6317–6322.
[32] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 18661–18673.
[33] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: International Conference on Machine Learning, PMLR, 2020, pp. 9929–9939.
[34] D. Kirange, R.R. Deshmukh, M. Kirange, Aspect based sentiment analysis semeval-2014 task 4, Asian J. Comput. Sci. Inf. Technol. 4 (8) (2014) 72–75.
[35] Q. Liu, Z. Gao, B. Liu, Y. Zhang, Automated rule selection for aspect extraction in opinion mining, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[36] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[37] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, 2008, pp. 231–240.
[38] L. Shu, H. Xu, B. Liu, Lifelong learning CRF for supervised aspect extraction, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017, pp. 148–154.
[39] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[40] Z. Ke, B. Liu, H. Wang, L. Shu, Continual learning with knowledge transfer for sentiment classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 683–698.
[41] G. Lv, S. Wang, B. Liu, E. Chen, K. Zhang, Sentiment classification by leveraging the shared knowledge from a sequence of domains, in: International Conference on Database Systems for Advanced Applications, Springer, 2019, pp. 795–811.
[42] J. Serra, D. Suris, M. Miron, A. Karatzoglou, Overcoming catastrophic forgetting with hard attention to the task, in: International Conference on Machine Learning, PMLR, 2018, pp. 4548–4557.
[43] H. Ahn, S. Cha, D. Lee, T. Moon, Uncertainty-based continual learning with adaptive regularization, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 32, Curran Associates, Inc., 2019.
[44] G. Zeng, Y. Chen, B. Cui, S. Yu, Continual learning of context-dependent processing in neural networks, Nat. Mach. Intell. 1 (8) (2019) 364–372.
[45] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 15920–15930.
[46] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751, http://dx.doi.org/10.3115/v1/D14-1181, URL https://aclanthology.org/D14-1181.
[47] J. von Oswald, C. Henning, B.F. Grewe, J. Sacramento, Continual learning with hypernetworks, in: International Conference on Learning Representations, 2019.

Qiangpu Chen is currently pursuing the Ph.D. degree in Sun Yat-Sen University, Guangzhou, China. He received his B.S. and MA.Eng degrees from the School of Software Engineering, Chongqing University, Chongqing, China, in 2014 and 2017, respectively. His research interests include foundations of deep learning, low-level computer vision, parallel computing, deep learning compiler, and resource allocation optimization.



Jiahua Huang received his MA.Eng degree from Sun Yat-Sen University, Guangzhou, China, in 2022. His research interests include deep learning, natural language processing, aspect-based sentiment analysis.

Wushao Wen is a professor in the School of Computer Science and Engineering, Sun Yat-Sen University, P.R. China. He received the B.S. degree from the University of Science and Technology of China in 1993, the M.S. and Ph.D. degrees from the University of California, Davis, in 1999 and 2001, respectively. He was an Engineer and Project Manager with China Telecommunication, Inc., from 1993 to 1997. He held various leading engineer and technical management positions in Cisco System, Ciena Corporation, Juniper Networks from 2000 to 2010 in Silicon Valley, USA, all in networking and security areas. He was appointed as the Chief Director of the Networks and Information Center, the Director of Shared Experimentation Teaching Center, Sun Yat-Sen University, P.R. China in 2013, 2014 respectively. He is currently doing research in cloud computing, network security, network architectures, machine learning, and deep learning.

Qingling Li is currently pursuing the Ph.D. degree in Sun Yat-Sen University, Guangzhou, China. She received his B.S. and MA.Eng degrees from the School of Computer Science, Shaanxi Normal University, Xi'an, China, in 2017 and 2019, respectively. Her research interests include foundations of deep learning, natural language processing, aspect-based sentiment analysis.

Rumin Zhang received his Ph.D.degree in Microelectronic and Solid electronics from Beihang University, China, in 2015. He is currently working as a professor of engineering at Ningbo Institute of Digital Twin (EIAS) and his research interests in deep learning, multimodal, AI compiler and so on.

Jinghui Qin is a lecturer at the Guangdong University of Technology, Guangzhou, China. He received his B.S. and MA.Eng degrees from the School of Software, Sun Yat-Sen University, Guangzhou, China, in 2012 and 2014, respectively. He also received his Ph.D degree from the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China, in 2020. He was a Post-Doctoral Fellow at the Sun Yat-sen University, Guangzhou, China, during 2020 and 2022. He has been serving as a reviewer of IEEE Trans. Neural Networks and Learning Systems, IEEE Trans. Broadcasting, Neurocomputing, Computer Vision and Image Understanding, Bioinformatics, etc. His research interest includes natural language processing, machine learning, and computer vision.