

# Bert adapter and contrastive learning for continual classification of aspect sentiment task sequences

Pham Thi Quynh Trang, Phan Dinh Dan Truong, Ngo Ngoc Huyen,  
Dang Thanh Hai\*

*Vietnam National University, University of Engineering and Technology,  
144 Xuan Thuy Street, Cau Giay District, Ha Noi, Viet Nam*

\*Email: [hai.dang@vnu.edu.vn](mailto:hai.dang@vnu.edu.vn)

Received: 31 July 2022; Accepted for publication: 5 January 2024

**Abstract.** Task incremental learning, a setting of continual learning, is an approach to exploit the knowledge from previous tasks for currently new tasks. Task incremental learning aims to solve two big challenges of continual learning: catastrophic forgetting and knowledge transfer or sharing between previous tasks and the current task. This paper improves task incremental learning by (1) transferring the knowledge (not the training data) learned from previous tasks to a new coming task (in contrast to multi-task learning); (2) maintaining or even improving the performance of learned models for previous tasks to avoid forgetting; (3) developing a continual learning model based on the results from (1) and (2) for aspect sentiment classification. Specifically, we combine two loss functions based on two contrastive learning modules, which are the Contrastive Knowledge Sharing (CKS) module for encouraging knowledge sharing between old and current tasks and the Contrastive Supervised learning (CSC) module for improving the performance of the current task. The experimental results show that our method could help previously learned tasks to get rid of the catastrophic forgetting phenomenon, outperforming previous studies for aspect sentiment classification.

**Keywords:** continual learning, catastrophic forgetting, knowledge transfer, contrastive learning, aspect sentiment classification.

**Classification numbers:** 4.7.4, 4.8.3

## 1. INTRODUCTION

Continual Learning (CL) (or lifelong learning) is defined as adaptive algorithms capable of learning from a continuous stream of information [1], where the information is progressively available over time and the number of learning tasks is not pre-defined. This learning setting is useful when the data privacy is a concern, i.e. the data owners do not want their data to be used by others. CL is aimed at leveraging the knowledge learned in the past to improve the new coming task learning performance.

There are three types of setting for CL: Class Incremental Learning (CIL), Task Incremental Learning (TIL) and Domain Incremental Learning (DIL) [2]. CIL contains non-overlapping classes and only one model is built for all classes. In the test phase, it does not know which task it is working with. TIL builds one model for each task, thus, when testing the task is

known. DIL only differs from TIL at this point, when testing the task is unknown. Continual learning has been successfully employed for building various aspect sentiment classification models. Given an example, if we consider "The mic quality" as the aspect of a mobile phone, and the sentence "the mic quality is quite nice" should be classified as "Positive" (instead of "Negative" or "Neutral") opinion about the studied aspect by ASC models. As we see, ASC only considers a number of pre-defined classes for all tasks, e.g. Positive, Negative and Neutral. Previous continual learning methods proposed for ASC are mainly based on fine-tuned BERT [3] over training data [3, 4]. However, some experiments, including ours show that this approach causes catastrophic forgetting for previous learned tasks, because the fine-tuned BERT on a task's training data set captures a highly task specific features that are likely not to be useful for others [5-8].

Inspired by the work of [9] (B-CL model), we exert the idea of Adapter-BERT [10] and further employ the continual learning adapters (CLA) rather than adapters in Adapter-BERT to avoid BERT parameters changing. We also use contrastive learning [11-13] that enables both knowledge transfer across tasks and knowledge distillation from previous tasks to the new task, eliminating the need for task identification in testing phase. In summary, this paper has two key contributions: (1) We propose to improve performance of continual learning for aspect sentiment classification (ASC) by integrating a Continual Learning Adapter (CLA) with contrastive learning loss. (2) We did extensive experiments on benchmark aspect sentiment classification data sets, demonstrating the efficiency of our proposal.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces our proposed method. Section 4 shows the experimental results. The final section is conclusions and future work.

## 2. RELATED WORK

Previous studies on continual learning often focus on solving the problem of catastrophic forgetting for learned tasks by the application of Contrastive Learning and a Bert-Adapter module.

### 2.1. Bert-Adapter module

There are several methods to take advantage of knowledge learned from pre-trained models, e.g. BERT, by fine-tuning or building an adapter module. With the fine-tuning method, models need to change the learned parameters to fit a new coming task. This approach takes a lot of time, computational resources and may suffer from catastrophic forgetting. With the other method, we just need to build an adapter module to be trained together with normalization layers, without any changes to any other BERT parameters [9]. As a result, this approach is suitable for CL since fine-tuning BERT itself causes serious forgetting. Within each transformer layer of BERT, Adapter-BERT simply inserts a 2-layer fully connected network (adapter). Adapter-BERT produces results that are comparable to fine-tuned BERT. Recently, a Network of Capsules (or Capsule Network) is a new classification neural architecture proposed by [14] and [15]. Unlike Convolutional Neural Networks (CNN), Capsule Networks (CapsNet) use vector capsules instead of scalar feature detectors to maintain additional valuable information, such as positions and thickness in an image. There are two capsule layers in a normal CapsNet. The primary layer contains low-level feature mapping, whereas the class layer generates classification probabilities, in which each capsule represents one class. It employs a dynamic

routing technique to allow each lower-level capsule to deliver its output to upper-level similar (or "agreed" as defined by the dot product) capsules.

## 2.2. Contrastive Learning

The goal of contrastive representation learning is to create an embedding space in which similar sample pairs are close to each other, whereas dissimilar sample pairs are far away. Both supervised and unsupervised settings can benefit from contrastive learning. It is one of the most potential ways in self-supervised learning when working with unsupervised data.

## 3. METHODS

Catastrophic forgetting and knowledge transfer are still two big challenges of continual learning. In this work we propose a continual learning model that is based on the CLASSIC model [16], a well-established continual learning model. We search and mask important neurons for each old task, then the model cannot change these neurons when training for new tasks through the task specific module (TSM). As a result, it helps the model mitigate catastrophic forgetting. In addition, to encourage knowledge transfer we identify and share knowledge from similar tasks to the current task through the knowledge sharing module (KSM). The TSM and KSM are two essential components of the continual learning adapter (CLA) in our model. However, different from the CLASSIC, we further improve by integrating contrastive supervised learning [12, 17] into the current task model (CSC) module. To this end, it will help our model enhance transferring knowledge learned from previous tasks to the current task, thus improving its performance. More details of each step will be presented as below.

### 3.1. Continual Learning Adapter

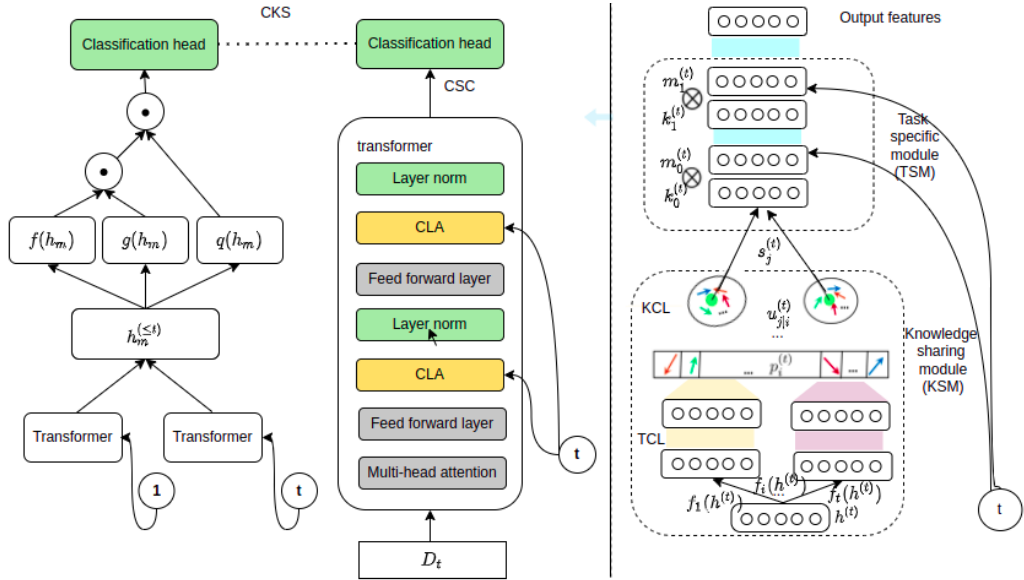


Figure 1. General architecture of our proposed method. On the left-hand side is about the CKS and CSC losses. The CKS is computed based on previous and current tasks and a task-based self-attention. CSC is computed based on the current task model. On the right-hand side is the overall architecture of CLA.

According to [16], CLA contains two modules: (1) Knowledge Sharing Module (KSM) for identifying and exploiting shareable knowledge from similar previous tasks and the new task, and (2) Task Specific Module (TSM) for learning task specific neurons and protecting them from being updated by the new coming task.

**Knowledge Sharing Module (KSM)** of CLA takes two inputs: (1) hidden states  $h(t)$  from the feed-forward layer inside a transformer layer and (2) task ID  $t$ . The outputs are hidden states with informative features for the  $t^{th}$  task. The KSM uses two capsule layers (task capsule and knowledge sharing capsule layers) and a dynamic routing algorithm to group similar tasks and shared knowledge (i.e. features) among tasks to enable knowledge transfer among similar tasks.

**Task Specific Module (TSM)** preserves task-specific knowledge (about the previous tasks) for preventing catastrophic forgetting by employing task masks (Figure 1). Specially, the TSM first detects the neurons used by old tasks, then masks out all used neurons when learning a new coming task. The task-specific module consists of differentiable layers (Note that CLA uses a 2-layer fully connected network). Each layer’s output is further applied with a task mask to indicate which neurons should be protected for that task, thus overcoming catastrophic forgetting and forbidding gradient updates for those neurons during back-propagation for a new task.

For task ID  $t$ , we denoted  $e_l^{(t)}$  as its embedding in layer  $l^{th}$  of the adapter, consisting of differentiable deterministic parameters that can be learned together with other parts of the network. It is trained for each layer in Task Specific Module (TSM). To generate the task mask  $m_l^{(t)}$  (a “soft” binary mask - is trained for each task  $t$  at each layer  $l$  in the adapter) from  $e_l^{(t)}$ , Sigmoid is used as a pseudo-gate function and a positive scaling hyper-parameter  $s$  is applied for training. The  $m_l^{(t)}$  is computed as follows:  $m_l^{(t)} = \sigma(se_l^{(t)})$

Note that neurons in  $m_l^{(t)}$  may overlap with those in others  $m_l^{(prev)}$  from previous tasks, which have some shared knowledge. Given the output task  $t$  of each layer adapter  $l^{th}$  in TSM denoted by  $k_l^{(t)}$ , we do element-wise multiplication  $k_l^{(t)} \otimes m_l^{(t)}$ . The masked output of the last layer  $k^{(t)}$  is fed into the next layer of the BERT with a skip connection (see Figure 1). After learning the task  $t$ , the final  $m_l^{(t)}$  is saved and added to the set  $\{m_l^{(t)}\}$ .

### 3.2. Contrastive learning on classification head

Inspired by contrastive learning and the CLASSIC model [16], we inject contrastive learning into two continual learning modules within CLASSIC to support our objective: contrastive knowledge sharing (CKS) to facilitate knowledge transfer, contrastive supervised learning on the current task model (CSC) to improve the current task model performance.

**Contrastive Knowledge Sharing (CKS)** aims to capture the shared knowledge among tasks and to help a new task learn a better representation for a better classifier. Intuitively, the more similar the two tasks are, the more shared knowledge they have. We, thus, use a task-based self-attention mechanism in our proposed model. We first transform the outputs of Adapter-BERT to other spaces via  $f(\cdot)$  and  $g(\cdot)$ . The similarity between two tasks  $i$  and  $j$  ( $i, j < t$ ) is calculated by:

$$s_{ij} = f(h_m^{(i)})^T g(h_m^{(j)}) \quad (1)$$

$h_m$  is the hidden state of the adapter after utilizing the task mask. Next, we compute the attention score  $\alpha_{i,j}$  to identify the tasks similar to the current task  $t$ :

$$\alpha_{i,j} = \frac{\exp(s_{i,j})}{\sum_j \exp(s_{i,j})} \quad (2)$$

Finally, we multiply the output of the attention layer with a scale parameter and add it back to the input feature  $h_m^{(\leq t)}$ :

$$h_{CKS}^{(\leq t)} = \sum_{i=1}^t (\gamma o_i + h_m^{(i)}) \quad (3)$$

After calculating the knowledge sharing view, we achieve two views: the output of current task  $h_m^{(t)}$  and the knowledge sharing view  $h_{CKS}^{(\leq t)}$  so that we can easily perform contrastive learning between them to encourage knowledge sharing. The contrastive loss between them is calculated as follows:

$$L_{CKS} = \sum_{i=1}^N - \frac{1}{N_{y_n}-1} \sum_{j=1}^N \frac{\delta_{n \leq j} \delta_{y_n=y_j} \log \left( \exp \left( \frac{h_{CKS:n}^{(\leq t)} \cdot h_{m:j}^{(t)}}{\tau} \right) \right)}{\sum_{k=1}^N \delta_{n \neq k} \exp \left( \frac{h_{CKS:n}^{(\leq t)} \cdot h_{m:k}^{(t)}}{\tau} \right)} \quad (4)$$

where  $N$  is the batch size and  $N_{y_n}$  is the number of examples in the batch that have the label  $y_n$ .  $h_{CKS:n}^{(\leq t)}$  and  $h_{m:k}^{(t)}$  corresponds to the hidden state of  $n^{th}$  sample in batch data after feeding to CKS module and adapter, respectively.

**Contrastive Supervised learning on the current task model** is exerted to improve the performance of the current task. To this end, we use the supervised contrastive loss:

$$L_{CSC} = \sum_{i=1}^N - \frac{1}{N_{y_n}-1} \sum_{j=1}^N \frac{\delta_{n \leq j} \delta_{y_n=y_j} \log \left( \exp \left( \frac{h_{m:n}^{(t)} \cdot h_{m:j}^{(t)}}{\tau} \right) \right)}{\sum_{k=1}^N \delta_{n \neq k} \exp \left( \frac{h_{m:n}^{(t)} \cdot h_{m:k}^{(t)}}{\tau} \right)} \quad (5)$$

where  $h_{m:j}^{(t)}$  is the hidden state of the  $j^{th}$  sample in the batch of task  $t$ .

**Total loss** or the final loss is the sum of three losses, including the cross entropy (CE), our two proposed CSC and CKS losses:

$$L = L_{CE} + \lambda_1 L_{CKS} + \lambda_2 L_{CSC}. \quad (6)$$

#### 4. EXPERIMENTAL RESULTS

We do extensive experiments of our proposed continual learning model for classification of aspect sentiments on the benchmark ASC data sets from 4 sources:

- (1) HL5Domains [18]: review sentences of 5 products;
- (2) Liu3Domains [19]: review sentences of 3 products;
- (3) Ding9Domains [20]: review sentences of 9 products; and
- (4) SemEval14 [21]: review sentences of 2 products.

To be consistent with the existing research, sentences with both positive and negative sentiments about one aspect are not used. In general, a single Aspect Sentiment Classification task is to classify whether a sentence expresses a positive, negative, or neutral opinion about a

given aspect. Formally, the objective of continual ASC is to accomplish a sequence of  $K$  ASC tasks  $\{T_1, T_2, \dots, T_K\}$  and an aspect term set  $R_K$  where the  $k^{th}$  task  $T_k$  has its own training set  $D_k$ . Suppose  $D_k$  contains  $N$  training samples  $\{(x_1, r_1, y_1), \dots, (x_N, r_N, y_N)\}$  where the instance  $(x_i, r_i, y_i), 1 \leq i \leq N$  indicates that the aspect term  $r_i$  in sentence  $x_i$  has the label as  $y_i \in \{positive, negative, neutral\}$ . A continual ASC model should perform well on ASC in all  $K$  tasks after being trained with the training data of these tasks coming sequentially.

We compare our proposed model with 3 continual learning based ASC models published very recently. These models are as follows:

- Non-continual learning approaches: fine-tune BERT and Bert-Adapter.
- Continual learning approaches: HAT [22] that focuses on solving catastrophic forgetting and B-CL [9] that is one of the most effective continual learning models.

We note that the running time for non-continual learning models is about 5 hours and for continual learning models is about 12 hours.

#### 4.1. Hardware configuration

The experiments were carried out on a machine with the following configuration.

- OS: Ubuntu 20.04
- CPU Intel Xeon CPU @ 2.30GHz
- RAM 13Gb
- GPU NVIDIA TESLA P100 for training.

#### 4.2. Datasets

Table 1. Number of review sentences in 10 datasets used in our experiments; each is considered a task in the context of continual learning.

Domain	Train	Validation	Test
Bing9domains_CanonS100	175	22	22
Bing9domains_ipod	153	19	20
Bing9domains_Nokia6600	362	45	46
Bing9domains_CanonPowerShotSD500	118	15	15
Bing5domains_CreativeLabs	677	85	85
Bing9domains_MicroMP3	484	61	61
Bing3domains_Speaker	352	44	44
Bing5domains_Nokia6610	271	34	34
Bing5domains_NikonCoolpix4300	162	20	21
Bing5domains_CanonG3	228	29	29

Because of limited computation resources, we randomly choose 10 datasets from 19 ASC datasets. The details of each chosen dataset are provided in Table 1. Each data set represents a

task in continual learning context and consists of a set of product review sentences annotated with sentiment for specific aspects. Sentences with both positive and negative sentiments on an aspect are ignored, which is the same as the work from [21].

#### 4.3. Hyperparameters

For each task-sharing module, we employ 2-layer fully connected network (dimension of 768) and 3 knowledge capsules. For each task-specific module, we use embeddings with the dimension of 2000 for the final and hidden layers of TSM. Each task id embedding has 2000 dimensions. In classification heads, we use softmax to evaluate the output. We use 5 epochs for each training; batch sizes for training and evaluation are 32 and 64, respectively; learning rate is set to 0.05. For  $\lambda_1$  and  $\lambda_2$ , we set the default to 1.

#### 4.4. Results and analysis

Table 2. Experimental ASC accuracy for 10 tasks from 10 models that exert Non-Continual Learning with fine-tuning Bert.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
95.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	90.0	91.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	90.0	91.3	93.3	0.0	0.0	0.0	0.0	0.0	0.0
95.5	90.0	91.3	93.3	87.1	0.0	0.0	0.0	0.0	0.0
95.5	90.0	91.3	93.3	87.1	75.4	0.0	0.0	0.0	0.0
95.5	90.0	91.3	93.3	87.1	75.4	90.9	0.0	0.0	0.0
95.5	90.0	91.3	93.3	87.1	75.4	90.9	91.2	0.0	0.0
95.5	90.0	91.3	93.3	87.1	75.4	90.9	91.2	100.0	0.0
95.5	90.0	91.3	93.3	87.1	75.4	90.9	91.2	100.0	72.4

Tables 2 and 3 show the results of the Non-Continual Learning models. As we can see, the performance on each task is not changed along the training progress.

The last row contains the final result of each model. Tables 4 and 5 show that the performance of most domains increases remarkably and is better than that of the HAT model. In particular, our model performs 7/10 tasks with significant performance improvement.

Table 6 shows that our model even does not need to be trained with previous tasks' data but still reaches equivalent performance. Furthermore, the performance on some domains is sustainably increased. For example, with Task 9, it begins with an accuracy of 75.9 %, then increases significantly to 85 % due to knowledge distillation from the former training.

Comparing our model with B-CL, 7/10 domains have higher performance, 2 have equal results as these two domains can reach an accuracy of 100 %.

Table 3. Experimental ASC accuracy for 10 tasks from 10 models that exert Non-Continual Learning with Bert-Adapter.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
75.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75.5	82.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75.5	82.5	91.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75.5	82.5	91.3	93.3	0.0	0.0	0.0	0.0	0.0	0.0
75.5	82.5	91.3	93.3	90.0	0.0	0.0	0.0	0.0	0.0
75.5	82.5	91.3	93.3	90.0	74.4	0.0	0.0	0.0	0.0
75.5	82.5	91.3	93.3	90.0	74.4	84.8	0.0	0.0	0.0
75.5	82.5	91.3	93.3	90.0	74.4	84.8	91.7	0.0	0.0
75.5	82.5	91.3	93.3	90.0	74.4	84.8	91.7	80.0	0.0
75.5	82.5	91.3	93.3	90.0	74.4	84.8	91.7	80.0	91.8

Table 4. Experimental ASC accuracy for 10 tasks from HAT continual learning model [22].

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
95.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	80.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	75.0	84.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	75.0	82.6	93.3	0.0	0.0	0.0	0.0	0.0	0.0
95.5	75.0	87.0	86.7	84.7	0.0	0.0	0.0	0.0	0.0
95.5	75.0	87.0	86.7	89.4	83.6	0.0	0.0	0.0	0.0
95.5	75.0	87.0	86.7	88.2	85.2	86.4	0.0	0.0	0.0
95.5	75.0	93.3	86.7	88.2	85.2	86.4	91.2	0.0	0.0
95.5	70.0	84.8	86.7	87.1	85.2	86.4	91.2	95.2	0.0
95.5	75.0	87.0	86.7	89.4	83.6	86.4	91.2	95.2	96.6

Table 5. Experimental ASC accuracy for 10 tasks from B-CL continual learning model [9].

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
95.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
90.9	79.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	79.0	92.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	79.1	92.9	83.9	0.0	0.0	0.0	0.0	0.0	0.0
90.9	76.0	88.2	74.2	93.2	0.0	0.0	0.0	0.0	0.0
90.9	75.4	88.2	74.2	95.5	94.1	0.0	0.0	0.0	0.0
90.9	74.6	87.1	74.2	93.2	97.1	100.0	0.0	0.0	0.0
81.8	67.7	82.4	74.2	93.2	94.1	100.0	100.0	0.0	0.0
86.4	75.4	84.7	80.6	93.2	94.1	100.0	100.0	75.9	0.0
90.9	77.4	88.2	90.3	100.0	94.1	100.0	100.0	85.0	96.6



Table 6. Experimental ASC accuracy for 10 tasks from our proposed continual learning model.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
95.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100.0	75.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	80.0	91.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95.5	80.0	93.5	100.0	0.0	0.0	0.0	0.0	0.0	0.0
100.0	80.0	91.3	93.3	92.0	0.0	0.0	0.0	0.0	0.0
95.5	80.0	95.7	100.0	90.6	86.9	0.0	0.0	0.0	0.0
95.5	75.0	93.5	100.0	88.2	85.2	93.2	0.0	0.0	0.0
95.5	75.0	93.5	100.0	87.1	85.2	93.2	97.1	0.0	0.0
90.9	70.0	93.5	100.0	84.7	88.5	93.2	97.1	100.0	0.0
100.0	75.0	95.7	100.0	87.1	85.2	93.2	97.1	100.0	82.8

We compare the results of our method with 4 baselines. Table 7 shows that our model has the highest average accuracy of 91.67 % across all 10 tasks, 0.92 % higher than that of BCL, 4.04 % higher than that of HAT. Likewise, when compared with non-continuous models, our model is much better, specifically 2.94 % higher than the fine-tune Bert based model and 6.1 % higher than the Bert-adapter based model.

Table 7. Comparison of our proposed model with some baseline models on ASC problem.

Model	Non-CL Fine-tune Bert	Non-CL Bert-Adapter	HAT	BCL	BCL+Contrastive
Task 1	95.5	75.5	90.5	90.9	<b>100.0</b>
Task 2	90.0	82.6	<b>95.5</b>	77.4	75.0
Task 3	91.3	88.9	72.9	88.2	<b>95.7</b>
Task 4	93.3	95.5	91.7	90.3	<b>100.0</b>
Task 5	87.1	90.0	<b>93.3</b>	100	87.1
Task 6	75.4	74.40	<b>94.1</b>	94.1	85.2
Task 7	90.9	84.8	90.9	<b>100.0</b>	93.2
Task 8	91.2	91.7	83.9	<b>100.0</b>	97.1
Task 9	100.0	80.0	77.9	74.1	<b>100.0</b>
Task 10	72.4	<b>91.8</b>	85.9	91.7	82.8
Avg	88.71	85.5	87.6	90.7	<b>91.6</b>

## 5. CONCLUSIONS AND FUTURE WORK

This paper proposes a solution to improve Continual Learning model's performance, demonstrating its power for a sequence of Aspect Sentiment Classification tasks. In particular, we study contrastive learning and the mechanism to adapt effective BERT in continual learning to solve 2 main problems of continual learning: catastrophic forgetting and knowledge sharing.

The experimental results illustrate that when we combine contrastive learning into B-CL, it can enhance the performance in the first train and help the model better prevent catastrophic forgetting than B-CL. On the other hand, when the model encounters new coming tasks that are not like the old tasks, the model's performance on the old tasks is unstable.

For future work, we plan to apply advanced Continual Learning techniques like Replay-Based approaches to handle this problem, improving the system performance. Another direction is to apply this proposal to other classification problems.

**Acknowledgements.** We would like to sincerely thank Assoc. Prof. Ha Quang Thuy from VNU University of Engineering and Technology for his great support, scientific advices, suggestions, and encouragement.

**CRedit authorship contribution statement.** Pham Thi Quynh Trang: Conceptualization, Methodology, Formal analysis, Investigation, Experiment verification, Supervision, Manuscript writing, edition and approval. Phan Dinh Dan Truong: Programming, Experiment conduction and investigation, Manuscript approval. Ngo Ngoc Huyen: Programming, Experiment conduction and investigation, Manuscript writing and approval. Dang Thanh Hai: Methodology, Formal analysis, Supervision, Manuscript edition and approval.

**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.

## REFERENCES

1. Parisi G. I., Kemker R., Part J. L., Kanan C., and Wermter S. - Continual lifelong learning with neural networks: A review, *Neural Networks* **113** (2019) 54-71. <https://doi.org/10.1016/j.neunet.2019.01.012>
2. Van de Ven G. M. and Tolias A. S. - Three scenarios for continual learning, arXiv preprint arXiv:1904.07734 (2019).
3. Devlin J., Chang M. W., Lee K., and Toutanova K. - BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
4. Xu H., Liu B., Shu L., and Yu P. S. - BERT post-training for review reading comprehension and aspect-based sentiment analysis, arXiv preprint arXiv:1904.02232 (2019).
5. Chen Z. and Liu B. - Lifelong machine learning, 2nd Ed., Morgan & Claypool Publishers, San Rafael, 2018.
6. Ke Z., Liu B., and Huang X. - Continual learning of a mixed sequence of similar and dissimilar tasks, *Advances in Neural Information Processing Systems* **33** (2020) 18493-18504.
7. Biesialska M., Biesialska K., and Costa-Jussa M. R. - Continual lifelong learning in natural language processing: A survey, arXiv preprint arXiv:2012.09823 (2020).
8. Huang Y., Zhang Y., Chen J., Wang X., and Yang D. - Continual learning for text classification with information disentanglement based regularization, arXiv preprint arXiv:2104.05489 (2021).
9. Ke Z., Xu H., and Liu B. - Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks, arXiv preprint arXiv:2112.03271 (2021).

10. Houshy N., Giurciu A., Jastrzebski S., Morrone B., De Laroussilhe Q., Gesmundo A., Attariyan M., and Gelly S. - Parameter-efficient transfer learning for NLP, Proceedings of the 36th International Conference on Machine Learning (ICML), Vol. 97, Long Beach, 2019, pp. 2790-2799.
11. Le-Khac P. H., Healy G., and Smeaton A. F. - Contrastive representation learning: A framework and review, IEEE Access **8** (2020) 193907-193934. <https://doi.org/10.1109/ACCESS.2020.3031549>
12. Khosla P., Teterwak P., Wang C., Sarna A., Tian Y., Isola P., Maschinot A., Liu C., and Krishnan D. - Supervised contrastive learning, Advances in Neural Information Processing Systems **33** (2020) 18661-18673.
13. Chen T., Kornblith S., Norouzi M., and Hinton G. - A simple framework for contrastive learning of visual representations, Proceedings of the 37th International Conference on Machine Learning (ICML), Vol. 119, Virtual Event, 2020, pp. 1597-1607.
14. Hinton G. E., Krizhevsky A., and Wang S. D. - Transforming auto-encoders, in: Honkela T., Duch W., Girolami M., Kaski S. (Eds.), Artificial Neural Networks and Machine Learning--ICANN 2011, Springer, Berlin, Heidelberg, 2011, pp. 44-51.
15. Sabour S., Frosst N., and Hinton G. E. - Dynamic routing between capsules, Advances in Neural Information Processing Systems **30** (2017) 3856-3866.
16. Ke Z., Liu B., Xu H., and Shu L. - CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks, arXiv preprint arXiv:2112.02714 (2021).
17. Jaiswal A., Babu A. R., Zadeh M. Z., Banerjee D., and Makedon F. - A survey on contrastive self-supervised learning, Technologies **9** (1) (2020) 2. <https://doi.org/10.3390/technologies9010002>
18. Hu M. and Liu B. - Mining and summarizing customer reviews, Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Seattle, 2004, pp. 168-177.
19. Liu Q., Gao Z., Liu B., and Zhang Y. - Automated rule selection for aspect extraction in opinion mining, Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, 2015, pp. 1291-1297.
20. Ding X., Liu B., and Yu P. S. - A holistic lexicon-based approach to opinion mining, Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM), Palo Alto, 2008, pp. 231-240.
21. Tang D., Qin B., and Liu T. - Aspect level sentiment classification with deep memory network, arXiv preprint arXiv:1605.08900 (2016).
22. Serra J., Suris D., Miron M., and Karatzoglou A. - Overcoming catastrophic forgetting with hard attention to the task, Proceedings of the 35th International Conference on Machine Learning (ICML), Vol. 80, Stockholm, 2018, pp. 4548-4557.