Boosting Large Language Models with Continual Learning for Aspect-based Sentiment Analysis

Anonymous ACL submission

Abstract

Aspect-based sentiment analysis (ABSA) is an important subtask of sentiment analysis, which aims to extract the aspects and predict their sentiments. Most existing studies focus on improving the performance of the target domain by fine-tuning domain-specific models (trained on source domains) based on the target domain dataset. Few works propose continual learning tasks for ABSA, which aim to learn the target domain's ability while maintaining the history domains' abilities. In this paper, we propose a Large Language Model-based Continual Learning (LLM-CL) model for ABSA. First, we design a domain knowledge decoupling module to learn a domain-invariant adapter and separate domain-variant adapters dependently with an orthogonal constraint. Then, we introduce a domain knowledge warmup strategy to align the representation between domain-invariant and domain-variant knowledge. In the test phase, we index the corresponding domain-variant knowledge via domain positioning to not require each sample's domain ID. Extensive experiments over 19 datasets indicate that our LLM-CL model obtains new state-of-the-art performance.

1 Introduction

Aspect-based sentiment analysis (ABSA) (Pontiki et al., 2016; Do et al., 2019; Zhang et al., 2022) plays an important role in the field of natural language processing. This task can be divided into two sub-tasks: aspect extract (AE), which aims to identify the aspects in the sentence and aspect-based sentiment classification (ABSC) (Zhou et al., 2019), which aims to infer the polarities of the corresponding aspects. For example, in the review “The service is bad but the food is delicious!", the user expresses negative and positive sentiments for aspects “service" and “food" respectively.

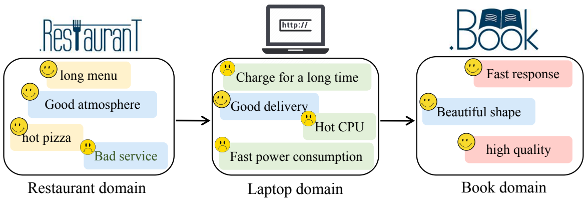


Figure 1: Continual learning for a sequence of ABSA domains. The blue color is domain-invariant knowledge, and the other is domain-variant knowledge.

The previous work for ABSA mainly trained a domain-specific model with designed architectures, which largely relies on the size of the target dataset (Li et al., 2018; Fei et al., 2022; Zhou et al., 2024). To utilize the datasets of other domains, transfer learning-based methods are proposed to learn the knowledge from source domains to the target domain (Marcacini et al., 2018; Zhou et al., 2021). However, these studies focus on improving the performance of the target domain, while ignoring the accuracy of source domains. To address this problem, a few studies introduced continual learning for a sequence of ABSA domains (Wang et al., 2018, 2020b; Ke et al., 2021c,d,a).

Wang et al. (2018) used a memory network to accumulate aspect sentiment knowledge by itself from big (past) unlabeled data and then used it to better guide its new/future task learning. Wang et al. (2020b) integrated a lifelong machine learning into Positive-Unlabeled (PU) learning model for target-based sentiment analysis. Ke et al. (2021c) introduced a novel contrastive continual learning method for knowledge transfer and distillation, and task masks to isolate task-specific knowledge to avoid catastrophic forgetting. To overcome catastrophic forgetting and transfer knowledge across domains, Ke et al. (2021d; 2021a) presented a novel capsule network based on pre-trained language models (e.g., BERT) to learn task-shared and task-specific knowledge via a masking strategy. They used a task-specific module for all the tasks, while the knowledge in different domains may conflict. Moreover, the relationships between the shared knowledge and specific knowledge are ignored by them.

There are still several challenges to continual learning for ABSA. **First (C1)**, this task requires rich commonsense knowledge to infer the sentiment. For example, the word “hot" expresses a negative sentiment polarity for the aspect “CPU" in the Laptop domain and has a positive sentiment for the aspect “pizza" in the Restaurant domain (See Figure 1). **Second (C2)**, the sentiment knowledge is inconsistent among different domains. The knowledge in each domain can be divided into domain-invariant knowledge (e.g., good, happy) and domain-variant knowledge (e.g., long, hot, fast). For instance, the general sentiment words are domain-invariant knowledge, which does not change among various domains.

To address these problems, we propose large language model-based continual learning (`LLM-CL`) for ABSA. Particularly, for **C1**, we integrate LLMs to utilize the large-scale commonsense knowledge in the model. Existing work has proved that LLMs can serve as a knowledge base (Petroni et al., 2019; Suchanek and Luu, 2023). Then, for **C2**, we individually consider the domain-invariant and domain-variant knowledge via a domain knowledge decoupling module with an orthogonal constraint. All the domains learn separate adapters for different domains with a shared adapter. Also, we propose a domain knowledge warmup mechanism to align the domain-invariant and -variant representation using replay data. In the test phase, we design a domain positioning strategy to index the correct domain-variant knowledge without knowing the domain the sample belongs to.

In the experiments, we first analyze the catastrophic forgetting problem of LLMs for ABSA. Although LLMs can reduce the catastrophic forgetting problem, it is still challenging for LLMs. Comparing our `LLM-CL` model on ABSC, AE, and JOINT tasks with several strong baselines, our model obtains new state-of-the-art performance on 19 datasets. The ablation studies show the effectiveness of the main components consisting of our `LLM-CL` model.

The key contributions are summarized as follows:

- We propose an LLMs-based CL framework for ABSA to leverage the rich commonsense knowledge in LLMs.
- We decouple domain-invariant and -variant knowledge by modeling the relationships among them using an orthogonal constraint. Then, a domain knowledge warmup strategy is proposed to align the representations of domain-invariant and -variant knowledge.
- We conduct extensive experiments on three subtasks over 19 domain datasets. The results show our `LLM-CL` model outperforms the existing typical baselines.

2 Related Work

2.1 Aspect-based Sentiment Analysis

Aspect-based sentiment analysis (ABSA) emerges as an advanced iteration of sentiment analysis, honing in on the intricate task of identifying specific aspects within a given text and subsequently extracting the associated polarity (Zhou et al., 2019). In this study, our focus is on its subtasks: aspect extraction (AE), which aims to pinpoint the aspects within a sentence, and aspect-based sentiment classification (ABSC), which seeks to deduce the polarities associated with the corresponding aspects. Neural network-based ABSA models designed domain-specific structures, such as attention (Wang et al., 2016), memory network (Tang et al., 2016), sequence to sequence (Yan et al., 2021) and graph neural network (Li et al., 2021; Wang et al., 2020a). All these models are based on large-scale labeled datasets, which is time-consuming and labor-intensive. Then, transfer learning is adopted for ABSA to transfer the knowledge from the source domain to the target domain (He et al., 2018), which focuses on improving the performance of the target domain.

2.2 Continual Learning for NLP

Continual learning (CL) is dedicated to acquiring new knowledge while addressing the prevalent issue of catastrophic forgetting, a subject extensively explored in NLP (Biesialska et al., 2020; Ke et al., 2023). Current research can be broadly categorized into three main approaches: rehearsal-based, regularization-based, and architecture-based methods. Rehearsal-based methods involve conducting experience replay by retaining historical information, which may take the form of preserved data (Li et al., 2022b; Scialom et al., 2022), or pseudo-data generators (Sun et al., 2019; Qin and Joty, 2022). Regularization-based methods enhance the loss function by introducing an additional term, commonly implemented through techniques such as knowledge distillation (Varshney et al., 2022;

Monaikul et al., 2021) or parameter importance (Li et al., 2022a; Liu et al., 2019). This modification aims to discourage alterations to crucial parameters acquired during a prior task when the model adapts to a new one. Architecture-based methods (Wang et al., 2023b,a; Razdaibiedina et al., 2023) allocate sets of task-specific parameters and dynamically integrate them with the frozen base model. These studies mainly focus on reducing the *catastrophic forgetting* problem based on *pre-trained language models* (e.g., BERT) whose parameters are much smaller than LLMs.

2.3 Continual Learning for ABSA

The most related works to our paper are (Ke et al., 2021c,d), which delved into the CL performance of pre-trained language models in ABSC. These works primarily designed a CL framework that performs well on the target domain while keeping the performance over the history domains. To overcome catastrophic forgetting, they shared a domain-specific module across all the domains and learned the domain-shared or domain-specific knowledge independently. However, domain-variant sentiment knowledge may conflict between the two domains. Moreover, domain-variant knowledge and domain-invariant knowledge are mutually exclusive with rich commonsense knowledge. Leveraging the capabilities of large language models, we model the relationships among domain-invariant and domain-variant knowledge and extend our investigation into ABSA, which performs AE and ABSC jointly.

3 Our Method

In this paper, we propose an LLMs-based CL framework for ABSA, which consists of domain knowledge decoupling, domain knowledge warmup and domain positioning (Figure 2). Our framework is based on an LLMs-based ABSA model, which trains a generative model using instruction tuning. We first introduce a domain knowledge decoupling module to learn a domain-invariant adapter with individual domain-variant adapters for each domain. Then, we align the domain-invariant and domain-variant representations via a domain knowledge warmup strategy. Finally, we utilize a domain positioning mechanism to index the domain-variant adapter without requiring the domain ID of each sample in the test stage.

Formally, given a sequence of domains $\{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^N\}$, we aim to sequentially learn a model f to maximize the function f at the domain $\mathcal{D}^i$ and history domains $\mathcal{D}^1, ..., \mathcal{D}^{i-1}$. Each domain $\mathcal{D}^i$ contains training samples $\{(x_j^i, y_j^i)\}_1^{|\mathcal{D}^i|}$, where (x_j^i, y_j^i) are the j-th example in domain domain $\mathcal{D}^i$, and $|\mathcal{D}^i|$ is the number of training samples in domain $\mathcal{D}^i$. Let x_j^i be a text in AE and JOINT or a text combined with a term in ABSC. Additionally let y_j^i be the aspect term in AE, or sentiment polarity (e.g., positive, negative and neutral) in ABSC or their combination in JOINT. Notably, in the test phase, we need to predict we randomly merge all the test samples selected from all domains without domain IDs.

3.1 LLMs-Based ABSA Model

Using a generative framework, we first build an LLMs-based ABSA model to integrate the rich latent knowledge in LLMs. We construct instructions to convert the input and output of ABSC and AE subtasks into a unified structure so that our model can perform all the tasks simultaneously.

Specifically, our instruction consists of input, prompt and output. The input is the sentence x_j^i we aim to predict. In prompt, we define the task (i.e., “Given a Sentence, you should extract all aspect terms and give a corresponding polarity") and the form of the output (i.e., “The format is "terms1: polarity1; terms2: polarity2"). In this way, the model can better understand the task and generate the response in a fixed format. As described in the prompt, we use “:" combine the aspect term and its polarity and use “;" combine multiple aspects in y_j^i.

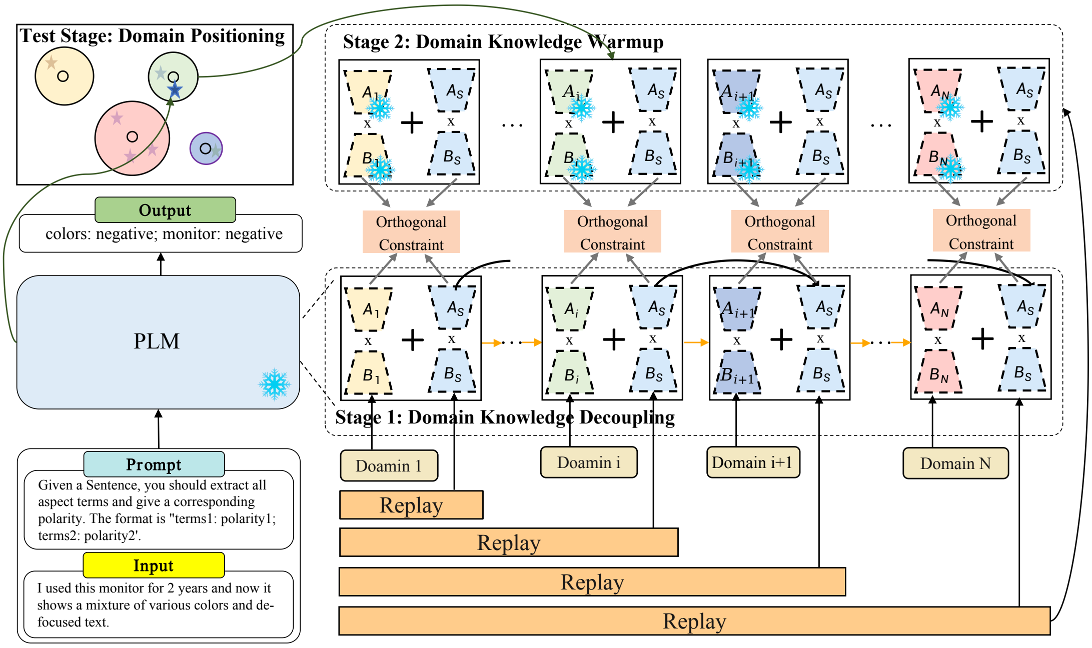
We adopt a parameter-efficient fine-tuning method, LoRA (Hu et al., 2022), which learns a low-rank adapter for each domain. The training objective is computed as follows:

$$f(y_j^i|x_j^i) = \mathrm{LLM}_{\phi+\theta}(\mathrm{output}_{y_j^i}|\mathrm{prompt}, \mathrm{input}_{x_j^i}) \tag{1}$$

where ϕ is the frozen pre-trained weights and θ is the domain-specific parameter increment, which $\theta \ll \phi$. In particular, the forward pass for LoRA are as follows:

$$h = \phi h_0 + \theta h_0 = \phi h_0 + BAh_0 \tag{2}$$

where $\theta = BA$ is the parameters of up matrix $A \in \mathcal{R}^{r,d}$ and down matrix $B \in \mathcal{R}^{d,r}$, h_0 and h are the text representation before and after encoding. d and r are the dimension of hidden representation and rank, where $d \ll r$.



Given a Sentence, you should extract all aspect terms and give a corresponding polarity. The format is "terms1: polarity1; terms2: polarity2'.

I used this monitor for 2 years and now it shows a mixture of various colors and de-focused text.

Figure 2: The framework of our LLM-CL.

3.2 Domain Knowledge Decoupling

Unlike the traditional LoRA model, we design a domain knowledge decoupling module to learn a domain-invariant adapter with separate domain-variant adapters. For the i-th domain, the training data including $\mathcal{D}^i$ and the replay data $\mathcal{D}^{R,i} = \{\mathcal{D}_1^R, ..., \mathcal{D}_k^R, ..., \mathcal{D}_i^R\}$. $\mathcal{D}_k^R$ means few examples sampled from the domain $\mathcal{D}^k$. We train the domain-invariant adapter $\theta_S = B_S A_S$ based on the replay data $\mathcal{D}^{R,i}$ using language modeling loss (LML).

$$\mathcal{L}_S^i = \sum_{\mathcal{D}_k^R \in \mathcal{D}^{R,i}} \sum_{(x_j^k, y_j^k) \in \mathcal{D}_k^R} \mathrm{LML}(y_j^k, f(y_j^k|x_j^k)) \tag{3}$$

Then, we train the domain-variant adapter $\theta_i = B_i A_i$ for the i-th domain based on domain data $\mathcal{D}^i$.

$$\mathcal{L}_D^i = \sum_{(x_j^i, y_j^i) \in \mathcal{D}^i} \mathrm{LML}(y_j^i, f(y_j^i|x_j^i)) \tag{4}$$

Furthermore, we utilize an orthogonal constraint to enforce the model to learn the difference between domain-invariant and domain-variant knowledge. To make sure B_i and A_i is orthogonal to B_S and A_S, we need to constrains them with $B_i^T B_S = 0$ and $A_i^T A_S = 0$. The loss is calculated as follows:

$$\mathcal{L}_O = \parallel A_i^T A_S \parallel^2 + \parallel B_i^T B_S \parallel^2 \tag{5}$$

Thus, the final training loss for domain knowledge decoupling is $\mathcal{L} = \mathcal{L}_S^i + \mathcal{L}_D^i + \lambda \mathcal{L}_O$, where λ is a hyper-parameter.

3.3 Domain Knowledge Warmup

Since the domain-variant adapter remains static post-training on a specific dataset, and the domain-invariant adapter undergoes changes throughout the training process, combining the two adapters directly can result in mismatches in parameter distributions and subsequent performance degradation. To address this, we leverage the replay data to fine-tune the invariant adapter for each variant adapter with frozen variant adapters. Specifically, following the competition of training for the N-th domain, we obtain a set of domain-variant adapters $(B_1, A_1), (B_2, A_2), ..., (B_N, A_N)$, along with a domain-invariant adapter (B_S, A_S). We process with additional training by combining each (B_i, A_i) with (B_S, A_S) using replay data $\mathcal{D}^{R,N}$, which comprises samples collected from all domains. To maintain the specificity of each domain-variant adapter, we only fine-tune the domain-invariant adapter in the process. This approach ensures that the domain-invariant adapter aligns with the parameter distribution differences among the domain-variant adapters, ensuring the effectiveness of subsequent combinations between them.

3.4 Domain Positioning

In the test phase, we need to index the domain-variant adapter of the test sample without knowing the domain ID the sample belongs to. Thus, we design a domain prototype learning module to learn the representation of the domain. Then, a nearest domain indexing module is presented to find the corresponding domain-variant adapter.

Domain Prototype Learning. Upon entering the test stage, we acquire N domain-variant adapters and corresponding domain-invariant adapters. As we lack knowledge of the domain ID corresponding to each sample at this stage, a strategy is needed to select the appropriate domain-variant adapter. We introduce a Domain Prototype Learning module to learn the recognizable representation of different domains based on the training data. For each training sample x_j^i in domain $\mathcal{D}^i$, we first obtain the average of the last block's hidden representations of the LLM, $h(x_j^i)$. Then we calculate each domain's mean μ^i and a shared covariance Σ to represent the domain,

$$\mu^i = \frac{1}{|\mathcal{D}^i|} \sum_{(x_j^i, y_j^i) \in \mathcal{D}^i} h(x_j^i) \tag{6}$$

$$\Sigma = \sum_{i=1}^{N} \frac{1}{|\mathcal{D}^i|} \sum_{(x_j^i, y_j^i) \in \mathcal{D}^i} (h(x_j^i) - \mu^i)(h(x_j^i) - \mu^i)^T \tag{7}$$

Nearest Domain Indexing. For a test sample x, we select the most matching domain-variant adapter using Mahalanobis distance,

$$-(h(x) - \mu^i)^T \Sigma^{-1} (h(x) - \mu^i) \tag{8}$$

4 Experimental Setups

4.1 Datasets and Metrics

Datasets Following the previous works (Ke et al., 2021d,c), we use 19 ABSA datasets which include product reviews to construct sequences of tasks. It consists (1) HL5Domains (Hu and Liu, 2004) with reviews of 5 products; (2) Liu3Domains (Liu et al., 2015) with reviews of 3 products; (3) Ding9Domains (Ding et al., 2008) with reviews of 9 products; and (4) SemEval14, with reviews of 2 products.

Metrics Considering the order of the 19 tasks can influence the final result, we randomly choose and run 3 task sequences, averaging their results for robust evaluation. In the case of ABSC, we calculate both accuracy and Macro-F1. The inclusion of Macro-F1 is crucial as it helps mitigate biases introduced in accuracy by imbalanced class distributions. Additionally, we compute F1 scores in both AE and JOINT. Following (Ke et al., 2021c,d), we adopt Average performance as an important metric in continuous learning, which reflects the comprehensive performance of the model on new and old tasks.

4.2 Selected Baselines

We evaluate `LLM-CL` against 15 typical baseline methods, which can be divided into two parts, Pre-trained Language Models (PLMs)-based and LLMs-based methods.

For PLMs-based methods, we compare with the following 10 strong baselines:

- **KAN** (Ke et al., 2021b) learns mask to activate units, facilitating optimized learning for the current task.

- **SRK** (Lv et al., 2019) learns knowledge and feature embeddings separately, and integrates them through a gate.

- **EWC** (Kirkpatrick et al., 2017) uses a regularization term to limit excessive updates of important parameters.

- **UCL** (Ahn et al., 2019) introduces a method based on a conventional Bayesian online learning framework.

- **OWM** (Zeng et al., 2019) adapts the parameters along a direction orthogonal to the input space of previous tasks.

- **HAT** (Serra et al., 2018) learns and utilizes pathways within a base network based on the task ID to construct task-specific networks.

- **B-CL** (Ke et al., 2021d) proposes a novel capsule network-based model for continual learning.

- **LAMOL** (Sun et al., 2019) employs a training strategy that involves both current data and samples derived from pseudo experience replay based on GPT-2.

- **CTR** (Ke et al., 2021a) integrates continual learning plug-ins into BERT.

- **CLASSIC** (Ke et al., 2021c) employs a contrastive continual learning method, facilitating knowledge transfer and knowledge distillation across tasks.

We also select some LLMs-based continual learning methods, which are based on LLaMA:

- **SEQUENCE** (Gururangan et al., 2020) utilizes a set of fixed-size LoRA parameters trained on a sequence of tasks.

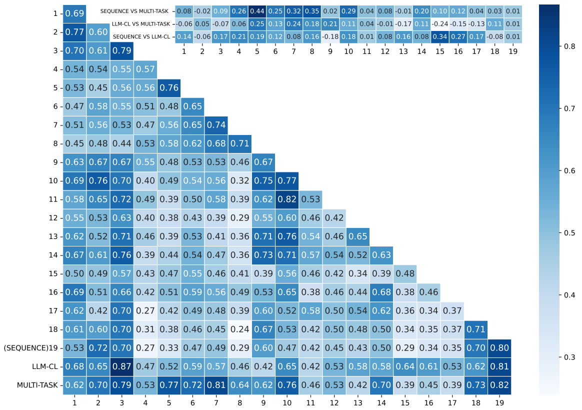


Figure 3: Catastrophic forgetting of LLM. The x-axis represents the test results for the corresponding domain. The y-axis represents the direction of the training domain from bottom to top. The subgraph in the upper right corner represents the gap between each method in each training domain. The depth of the color in the grid indicates how well the LLM performs on the corresponding test set during the continual learning process.

- **REPLAY** (Chaudhry et al., 2019) saves 8 samples of each previous task as memory and trains a fix-sized LoRA one step on the memory after every 5 steps of training on the new task. For a fair comparison, we also adopt the replay to O-LoRA and AdaLoRA.

- **O-LoRA** (Wang et al., 2023a) focuses on learning new tasks within an orthogonal subspace while maintaining fixed LoRA parameters for previously learned tasks.

- **AdaLoRA** (Zhang et al., 2023) adaptively allocates parameter budgets among weight matrices based on importance scores and parameterizes incremental updates using singular value decomposition.

- **Multi-task** (Caruana, 1997) trains a set of fixed-size LoRA parameters on all tasks as multi-task learning, which is the upper bound of continual learning.

4.3 Experimental Settings

In our experiment, we adopt LLaMA-7B (Touvron et al., 2023) as our base model. We train all models using AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ coupled with a cosine scheduler with the initial learning rate of $5e-5$. For all orders of task sequences, we trained the models with 30 epochs, a batch size of 16 on NVIDIA RTX 4090 with 24GB video memory. And we trained 10 epochs in domain knowledge warmup. We set the default LoRA rank to 8. For every domain, we randomly preserve 8 samples for replay. For domain knowledge decoupling and domain knowledge warmup, we set $\lambda = 1e-6, 1e-5$ separately.

5 Experimental Analysis

5.1 Catastrophic Forgetting of LLMs

In Figure 3, We explore the catastrophic forgetting problem of LLMs on ABSA. We find: (1) LLMs still meet the catastrophic forgetting problem. For example, the model trained on the first 8 domains obtains a 0.71 F1 score on the 8-th domain, while the model trained on the first 18 domains obtains only 0.24. (2) LLM-CL showcases its effectiveness in mitigating catastrophic forgetting. We observe that LLM-CL obtains improvement over SEQUENCE in most domains (16/18). While approaching the performance of the multi-task model, LLM-CL has even surpassed Multi-task in 7 domains.

5.2 Main Results

In Table 1, we compare our method with precious continual learning methods for ABSC and some LLMs-based continual learning methods. Additionally, we extend to more challenging ABSA subtasks, AE and JOINT in Table 2.

Peformance on ABSC. Overall, LLM-CL outperforms all baselines markedly. We also find: (1) SEQUENCE achieves comparable results to previous CL methods, which show the powerful performance of LLMs. (2) Compared to rehearsal-free CL methods, replaying a certain proportion of historical data can improve the CL methods in most cases. However, replay data can still affect the model's ability to cope with data requiring domain-specific knowledge. (3) Compared to the previous SOTA CL method for ABSC, CLASSIC, our method improves from 0.9022 to 0.9491 in Accuracy and from 0.8512 to 0.9143 in Macro-F1. Noteworthily, our method achieves results comparable to Multi-task in Accuracy and gets 4.38% improvement on Macro-F1, which shows our methods can extract shared and specific knowledge during continual learning settings, thereby mitigating catastrophic forgetting. Multi-task merges the datasets from multiple domains simply, where the inconsistent (domain-specific) knowledge may influence the performance.

		Order 1		Order 2		Order 3		Average	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
PLMs	KAN*	-	-	-	-	-	-	0.8549	0.7738
	SRK*	-	-	-	-	-	-	0.8476	0.7852
	EWC*	-	-	-	-	-	-	0.8637	0.7452
	UCL*	-	-	-	-	-	-	0.8389	0.7482
	OWM*	-	-	-	-	-	-	0.8702	0.7931
	HAT*	-	-	-	-	-	-	0.8674	0.7816
	B-CL*	-	-	-	-	-	-	0.8829	0.8140
	LAMOL*	-	-	-	-	-	-	0.8891	0.8059
	CTR*	-	-	-	-	-	-	0.8947	0.8362
	CLASSIC†	-	-	-	-	-	-	0.9022	0.8512
LLMs	SEQUENCE	0.8994	0.7215	0.9405	0.8895	0.9430	0.9017	0.9276	0.8376
	REPLAY	0.9212	0.7444	0.9367	0.8765	0.9377	0.8837	0.9319	0.8349
	O-LoRA	0.8822	0.6752	0.9429	0.8923	0.9400	0.8974	0.9217	0.8216
	O-LoRA$_{\text{replay}}$	0.9071	0.7897	0.9196	0.8421	0.9350	0.8300	0.9206	0.8206
	AdaLoRA	0.8553	0.6385	0.9332	0.8574	0.9227	0.8435	0.9037	0.7798
	AdaLoRA$_{\text{replay}}$	0.9086	0.7822	0.9387	0.8778	0.9269	0.8659	0.9247	0.8420
	LLaMA (0/4-shot)	-	-	-	-	-	-	-	-
	Alpaca (0/4-shot)	-	-	-	-	-	-	-	-
	GPT-3.5-Turbo (0-shot)	-	-	-	-	-	-	0.9098	0.7086
	GPT-3.5-Turbo (4-shot)	-	-	-	-	-	-	0.9269	0.6316
Ours	LLM-CL	**0.9498**	**0.9123**	**0.9495**	**0.9155**	**0.9480**	**0.9150**	**0.9491**	**0.9143**
Upper bound	Multi-task	-	-	-	-	-	-	0.9492	0.8705

Table 1: The main results on ABSC in terms of Accuracy (Acc.) and Macro-F1 (F1). * and † denote the results come from (Ke et al., 2021a) and (Ke et al., 2021c). The best results of all methods are bolded.

Peformance on AE and JOINT. The conclusions derived from Table 2 generally align with Table 1, and we also have some observations: (1) Our method has a more significant improvement in the capabilities of these two subtasks, while there is still a potential room for improvement compared with the upper bound. (2) In all subtasks, we find that O-LoRA and AdaLoRA, even with the addition of replay, did not achieve better results than REPLAY. We believe that these two methods mainly focus on the differences between different tasks while ignoring the shared knowledge between domains, which requires special attention in the continual learning for ABSA.

5.3 Ablation Studies

To further inspect our methods, we conduct analyses to investigate the effect of LLM-CL's components. Specifically, we investigate the effect of (1) - Orthogonal Constraint(- OC), in which we remove the constraint between the domain-invariant adapter and separate domain-variant adapter. (2) - Domain Knowledge Decoupling(- DKD), in which we merge two adapters directly without distinguishing them. (3) - Domain Knowledge Warmup(- DKW), in which we skip the stage of Domain Knowledge Warmup. (4) - Domain Positioning(- DP), which we replace with Random Positioning.

We observe the following findings: (1) Orthogonal constraint can effectively extract domain-variant knowledge that is orthogonal to invariant knowledge, which was more pronounced in more challenging subtasks such as AE and JOINT. (2) Simply decoupling the adapter has no advantage compared to the original adapter, while our method improves it due to considering the constraints between different adapters. (3) Unlike the ABSC, the domain-invariant adapter exhibits a heightened capacity for acquiring broader knowledge during continual learning across domains, particularly in the context of AE and JOINT tasks. The integration of Domain Knowledge Warmup further enhances its adaptability to the domain-variant adapter, where F1 has elevated from 0.5180 to 0.6785 on AE and from 0.3327 to 0.5867 on JOINT. (4) Utilizing Domain Positioning, our approach adeptly identifies the fitting domain-variant adapter for predictions. This underscores the discernible distinctions in data distribution across various fields, demonstrating the efficacy of LLMs' capabilities in leveraging these domain-variant characteristics.

5.4 Further Analysis

Comparison with SOTA LLMs. As LLMs demonstrate the capability to learn new tasks solely through natural language instructions, we investigate the performance of SOTA LLMs in 0-shot and few-shot scenarios. As shown in Table 1 and Table 2, we select LLaMA, Alpaca (instruction fine-tuned version of LLaMA) and GPT-3.5-Turbo us-

	AE				JOINT			
	Order 1	Order 2	Order 3	Average	Order 1	Order 2	Order 3	Average
SEQUENCE	0.6262	0.6003	0.6734	0.6333	0.4817	0.4939	0.5428	0.5061
REPLAY	0.6236	0.6684	0.6774	0.6565	0.5300	0.5309	0.5637	0.5415
O-LoRA	0.6116	0.6043	0.6849	0.6336	0.4507	0.4943	0.5751	0.5067
O-LoRA$_{\text{replay}}$	0.6077	0.6034	0.6633	0.6248	0.5286	0.5392	0.5564	0.5414
AdaLoRA	0.6007	0.5575	0.6376	0.5986	0.4213	0.4586	0.5079	0.4626
AdaLoRA$_{\text{replay}}$	0.6136	0.5818	0.6514	0.6156	0.4803	0.5278	0.5432	0.5171
LLaMA (0/4-shot)	-	-	-	-	-	-	-	-
Alpaca (0/4-shot)	-	-	-	-	-	-	-	-
GPT-3.5-Turbo (0-shot)	-	-	-	0.4663	-	-	-	0.3919
GPT-3.5-Turbo (4-shot)	-	-	-	0.5610	-	-	-	0.4886
LLM-CL (ours)	**0.6719**	**0.6758**	**0.6877**	**0.6785**	**0.5893**	**0.5829**	**0.5878**	**0.5867**
Upper bound (Multi-task)	-	-	-	0.7033	-	-	-	0.6235

Table 2: The F1 scores over AE and JOINT tasks.

	ABSC		AE	JOINT
	Acc.	F1	F1	F1
LLM-CL	**0.9491**	**0.9143**	**0.6785**	**0.5867**
- OC	0.9443	0.9050	0.6500	0.5676
- DKD	0.9334	0.8744	0.6630	0.5732
- DKW	0.9447	0.9054	0.5180	0.3327
- DP	0.9378	0.8846	0.6456	0.5207

Table 3: The results of ablation studies.

	ABSC		AE	JOINT	
r	Acc.	F1	F1	F1	Score
4	0.9460	0.9197	0.6818	0.5298	0.4882
8	0.9498	0.9123	0.6719	0.5893	0.7536
16	0.9465	0.9124	0.6865	0.5700	0.6996
32	0.9450	0.8812	0.6681	0.5697	0.1647

Table 4: Influence of rank r on ABSC (order 1).

ing 0-shot and 4-shot. We observed that 1) LLaMA and Alpaca fail to predict the answer both in two scenarios. This observation underscores the necessity of fine-tuning procedures for some LLMs, particularly when tackling intricate tasks like ABSA. 2) GPT-3.5-Turbo shows powerful 0-shot and 4-shot capabilities, but there is still a certain gap compared to the fine-tuned model.

The Influence of Rank r. Since our method is a variant of Lora, an important influencing factor is rank r. We study the hyperparameter sensitivity by setting rank r with values in [4, 8, 16, 32] for LLM-CL and conducted experiments on order1 of ABSC. We calculate $Score$ as follows:

$$Score = \frac{1}{|M|} \sum_{m \in M} \frac{p_{r,m} - min(p_{*,m})}{max(p_{*,m}) - min(p_{*,m})}$$

where M includes is a set of metrics on each subtask, $p_{i,j}$ represents the performance of LLM-CL on metric j when rank $r = i$.

As shown in Table 4, we find that with rank r increasing, $Score$ initially improves and then deteriorates, reaching its optimum when rank $r = 8$. This suggests, on one hand, that excessively small rank r can hinder the model's ability to effectively capture the diversity of tasks. On the other hand, overly large rank r may lead to overfitting.

5.5 Conclusions and Further Work

This paper introduces a novel approach, the LLMs-based continual learning framework, LLM-CL, designed for ABSA. It effectively separates domain-invariant and -variant knowledge by incorporating an orthogonal constraint to model their relationships. To bridge the gap between these knowledge types, we introduce a domain knowledge warmup strategy, which focuses on aligning representations of domain-invariant information. We observe that LLMs still have the problem of catastrophic forgetting despite obtaining great improvement compared with traditional models. Experiments show that LLM-CL markedly improves the performance on three subtasks over 19 domain datasets. In future work, we would like to explore the effectiveness of our model on other cross-domain continual learning tasks.

Limitations

Although the effectiveness of our method has been validated across the three subtasks of ABSA, there is still room for improvement. Firstly, our method decouples the traditional adapter into a domain-invariant adapter and a domain-variant adapter. However, as the number of domains increases, the storage requirements also grow. More fine-grained decoupling will be the focus of our future research. Secondly, during the test phase, additional inference is required for samples to obtain implicit information for domain positioning. To improve efficiency and performance, our method needs a more lightweight and efficient model for domain prototype learning. By addressing these limitations, we can enhance the scalability and performance of our method, further advancing the development of LLMs in CL.

References

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. *NeurIPS*, 32.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of COLING*, pages 6523–6541.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of WSDM*, pages 231–240.

Hai Ha Do, Penatiyana WC Prasad, Angelika Maag, and Abeer Alsadoon. 2019. Deep learning for aspect-based sentiment analysis: a comparative review. *Expert systems with applications*, 118:272–299.

Hao Fei, Fei Li, Chenliang Li, Shengqiong Wu, Jingye Li, and Donghong Ji. 2022. Inheriting the wisdom of predecessors: A multiplex cascade framework for unified aspect-based sentiment analysis. In *Proceedings of IJCAI*, pages 4096–4103.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. In *Proceedings of ACL*, pages 579–585.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021a. Achieving forgetting prevention and knowledge transfer in continual learning. *Proceedings of NeurIPS*, 34:22443–22456.

Zixuan Ke, Bing Liu, Hao Wang, and Lei Shu. 2021b. Continual learning with knowledge transfer for sentiment classification. In *ECML PKDD*, pages 683–698. Springer.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021c. Classic: Continual and contrastive learning of aspect sentiment classification tasks. In *Proceedings of EMNLP*, pages 6871–6883.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *Proceedings of ICLR*.

Zixuan Ke, Hu Xu, and Bing Liu. 2021d. Adapting bert for continual learning of a sequence of aspect sentiment classification tasks. *arXiv preprint arXiv:2112.03271*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. 2022a. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Proceedings of NAACL*, pages 5441–5454.

Guodun Li, Yuchen Zhai, Qianglong Chen, Xing Gao, Ji Zhang, and Yin Zhang. 2022b. Continual few-shot intent detection. In *Proceedings of COLING*, pages 333–343.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of ACL*, pages 6319–6329.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of ACL*, pages 946–956.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *IJCAI*.

Tianlin Liu, Lyle Ungar, and Joao Sedoc. 2019. Continual learning for sentence representations using conceptors. *arXiv preprint arXiv:1904.09187*.

Guangyi Lv, Shuai Wang, Bing Liu, Enhong Chen, and Kun Zhang. 2019. Sentiment classification by leveraging the shared knowledge from a sequence of domains. In *DASFAA*, pages 795–811. Springer.

Ricardo Marcondes Marcacini, Rafael Geraldeli Rossi, Ivone Penque Matsuno, and Solange Oliveira Rezende. 2018. Cross-domain aspect extraction for sentiment analysis: A transductive learning approach. *Decision Support Systems*, 114:70–80.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. In *Proceedings of AAAI*, volume 35, pages 13570–13577.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics.

Chengwei Qin and Shafiq Joty. 2022. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *ICLR*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *arXiv preprint arXiv:2205.12393*.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4548–4557. PMLR.

Fabian Suchanek and Anh Tuan Luu. 2023. Knowledge bases and language models: Complementing forces. In *International Joint Conference on Rules and Reasoning*, pages 3–15. Springer.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. *arXiv preprint arXiv:1909.03329*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaibhav Varshney, Mayur Patidar, Rajat Kumar, Lovekesh Vig, and Gautam Shroff. 2022. Prompt augmented generative replay via supervised contrastive learning for lifelong intent detection. In *Findings of NAACL*, pages 1113–1127.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020a. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of ACL*, pages 3229–3238.

Shuai Wang, Guangyi Lv, Sahisnu Mazumder, Geli Fei, and Bing Liu. 2018. Lifelong learning memory networks for aspect sentiment classification. In *Proceedings of Big Data*, pages 861–870. IEEE.

Shuai Wang, Mianwei Zhou, Sahisnu Mazumder, Bing Liu, and Yi Chang. 2020b. Disentangling aspect and opinion words in sentiment analysis using lifelong pu learning. In *New Frontiers in Mining Complex Patterns: 8th International Workshop, NFMCP 2019, Held in Conjunction with ECML-PKDD 2019, Würzburg, Germany, September 16, 2019, Revised Selected Papers 8*, pages 100–115. Springer.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023a. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of EMNLP*, pages 606–615.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023b. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of ACL*, pages 10933–10946.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of ACL*, pages 2416–2429.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. 2019. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Jie Zhou, Jimmy Xiangji Huang, Qin Chen, Qinmin Vivian Hu, Tingting Wang, and Liang He. 2019. Deep learning for aspect-level sentiment classification: Survey, vision, and challenges. *IEEE access*, 7:78454–78483.

Jie Zhou, Yuanbiao Lin, Qin Chen, Qi Zhang, Xuanjing Huang, and Liang He. 2024. Causalabsc: Causal inference for aspect debiasing in aspect-based sentiment classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:830–840.

Yan Zhou, Fuqing Zhu, Pu Song, Jizhong Han, Tao Guo, and Songlin Hu. 2021. An adaptive hybrid framework for cross-domain aspect-based sentiment analysis. In *Proceedings of AAAI*, volume 35, pages 14630–14637.