Multilingual Continual Learning using Attention Distillation

Sanjay Agrawal
Amazon, India
sanjagr@amazon.com

Deep Nayak
Amazon, India
deepnyk@amazon.com

Vivek Sembium
Amazon, India
viveksem@amazon.com

Abstract

Query-product relevance classification is crucial for e-commerce stores like Amazon, ensuring accurate search results that match customer intent. Using a unified multilingual model across multiple languages/marketplaces tends to yield superior outcomes but also presents challenges, especially in maintaining performance across all languages when the model is updated or expanded to include a new one. To tackle this, we examine a multilingual continual learning (CL) framework focused on relevance classification tasks and address the issue of catastrophic forgetting. We propose a novel continual learning approach called attention distillation, which sequentially adds adapters for each new language and incorporates a fusion layer above language-specific adapters. This fusion layer distills attention scores from the previously trained fusion layer, focusing on the older adapters. Additionally, translating a portion of the new language data into older ones supports backward knowledge transfer. Our method reduces trainable parameters by 80%, enhancing computational efficiency and enabling frequent updates, while achieving a 1-3% ROC-AUC improvement over single marketplace baselines and outperforming SOTA CL methods on proprietary and external datasets.

1 Introduction

Large-scale e-commerce search systems, like those of Amazon and Walmart, employ a multi-step process to retrieve relevant products. Initially, a product set approximating relevance to the query is generated (Agrawal et al., 2023b) (Agrawal et al., 2023a), followed by optimization steps for relevance, customer interest, and other metrics (Momma et al., 2022). Accurately capturing relevance between a customer's query-intent and the product set is crucial for a positive customer experience, leading to the adoption of relevance models in various marketplaces. However, as new marketplaces emerge, the need for language-specific relevance models arises, resulting in the maintenance of multiple models. Yet, achieving semantic alignment across languages and utilizing a single model trained on data from all marketplaces can enhance knowledge transfer (Zhang and Yang, 2021; Liu et al., 2019). However, creating a single model for multiple marketplaces presents challenges; expanding to new marketplaces demands retraining the entire model with data from all existing ones, incurring substantial computational costs and necessitating simultaneous access to data from all marketplaces during training. This paper addresses these challenges in a continual learning scenario, where marketplaces are introduced sequentially (see Figure 1). This scenario demands model updates to accommodate new marketplaces while preserving performance for older ones, without replaying data from older marketplaces. Please note that introducing a new marketplace implies the presence of data in a new language.

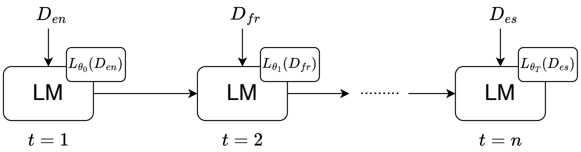

Figure 1: In Continual Learning, the model (LM) trains on one dataset at a time, starting with EN language, then FR language, and so on. Parameters are updated sequentially using loss function $L(.)$. This diagram demonstrates the CL training framework and is not a production representation.

In this context, we propose a novel approach called attention distillation, wherein adapters (Rebuffi et al., 2017a) are progressively incorporated for each marketplace data. In this context, an adapter fusion layer (Pfeiffer et al., 2020a) is incorporated with randomly initialised weights at every time

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 91–99
January 19–24, 2025. ©2025 Association for Computational Linguistics

step t that sits atop the adapters. In this case (see Figure 2(b)), the attention scores related to previous adapters in the new fusion layer are distilled from the previously trained fusion layer. While, the attention scores for the new marketplace adapter are trained using the conventional approach as detailed in (Pfeiffer et al., 2020a) for their specific new target tasks. Furthermore, we introduce utilizing a subset of new language data translated into the older language datasets to enable backward knowledge transfer through our proposed methodology. Our experimental focus addresses the following research questions: **RQ1**: Given Adapter fusion operates in a non-sequential manner, can our proposed approach attain similar performance in continual learning while also reducing a significant number of parameters? **RQ2**: How effective are state-of-the-art Continual Learning Methods in transferring knowledge in multilingual scenarios? **RQ3**: What is the impact of translating new marketplace language data at time t into the older marketplace languages within a continual learning setup on knowledge transfer when training for new marketplace? Our **key contributions** include:

1. We propose a novel attention distillation method tailored for continual learning: (a) We introduce an adapter fusion layer with randomly initialized weights at each time step t, positioned above the adapters. This layer distills attention scores related to previous adapters from the previously trained fusion layer. (b) Furthermore, we facilitate backward knowledge transfer by translating some new marketplace data into older ones, leveraging our attention distillation approach.

2. Empirical evaluation of the proposed approach on proprietary and public datasets results in a significant boost of 1-3% ROC-AUC on the query-product relevance task compared to training each marketplace dataset separately. Our approach also outperforms existing SOTA CL algorithms when evaluating relevance classification tasks across various languages within a continual learning context. our approach facilitates a significant reduction of trainable parameters in a transformer model—up to 80%—when expanding to new languages.

2 Problem Statement

We define the multi-lingual continual learning problem as follows: Consider n distinct marketplace datasets $\{D_1, D_2, \ldots, D_n\}$, each in a unique language. We train a multilingual transformer model sequentially on these datasets, excluding older data to improve computational efficiency. For example, when training on D_t, we exclude $\{D_{t-1}, D_{t-2}, \ldots, D_1\}$.

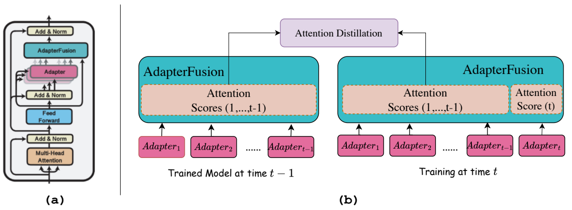

Figure 2: (a) AdapterFusion (Pfeiffer et al., 2020a) in a transformer takes inputs from various task-specific adapters, learning to mix their encoded information. (b) Our proposed method integrates attention distillation into a continual learning framework, conducting training at time t while leveraging knowledge from the previously trained model at time step $t-1$.

Let M_t represent the model trained on D_t, built upon M_{t-1}. Our goal is to fine-tune M_t using D_t while preserving performance on previous datasets $\{D_1, D_2, \ldots, D_{t-1}\}$ and mitigating catastrophic forgetting. The model parameters at time t are Θ_{M_t}, and the base transformer model parameters are Θ_{base}. The task-specific loss function for M_t is $\mathcal{L}_t$.

3 Related Work

Continual Learning: Continual learning methods generally fall into four categories: **(i) Replay-based methods:** These techniques involve caching a portion of data for each new task introduced to the model. The system then utilizes experience replay to prevent catastrophic forgetting, as illustrated in prior work by Dautume et al. (de Masson d'Autume et al., 2019) (Rebuffi et al., 2017b). **(ii) Regularization-based methods:** These apply regularization loss to various model components to prevent significant deviations from previously learned tasks. Regularization can target the output (Li and Hoiem, 2017), hidden space (Rannen et al., 2017), or model parameters (Lopez-Paz and Ranzato, 2022) (Zenke et al., 2017). **(iii) Architecture-based methods:** These methods design model segments to handle specific tasks, reducing interference between tasks. Examples include (Rusu et al., 2022) and (Mallya and Lazebnik, 2018), with our approach inspired by the CTR architecture (Ke et al., 2021a). **(iv) Meta-Learning based methods:** These focus on optimizing knowledge transfer across tasks (Riemer et al., 2019). For instance, (Wang et al., 2022) introduces prompt learning to adapt Large Language Models (LLMs) to new tasks.

Adapters and Adapter Fusion: Adapters are small parameter efficient fully connected networks that are introduced at every layer of a transformer model. In the work by (Pfeiffer et al., 2020a), Adapter Fusion is introduced as an attention layer placed on top of these Adapters. It's purpose is to encourage the non-destructive transfer of knowledge between various task-specific adapters shown in Figure 2(a).

Components of Fusion Layer: Adapter Fusion is trained to compose the n task-specific adapters $\{\Theta_1, ..., \Theta_n\}$ and the shared pretrained model Θ_{base} through the introduction of a new set of weights Ψ. As shown in Figure 2(a), we note that the AdapterFusion parameters Ψ encompass Key, Value, and Query matrices at each layer denoted as K_l, V_l, and Q_l, respectively. For each layer l of the transformer and at each token-step j, the output from the feedforward sub-layer of layer l serves as the query vector. The output of each adapter, $z_{l,j}$, is employed as input for both value and key transformations. As outlined in Vaswani et al. (Vaswani et al., 2017), we learn a contextual activation for each adapter t using

$$s_{l,j} = softmax(h_{l,j}^T Q_l \cdot z_{l,j,t}^T K_l), t\epsilon\{1, ..., n\} \quad (1)$$

$$z'_{l,j,t} = z_{l,j,t}^T V_l, t\epsilon\{1, ..., n\} \quad (2)$$

$$Z'_{l,j} = [z'_{l,j,1}, ..., z'_{l,j,n}] \quad (3)$$

$$o_{l,j} = s_{l,j}^T Z'_{l,j} \quad (4)$$

Here, n denotes the total count of adapters.

4 Proposed Methodology

This paper aims to develop a query-product relevance classification model (Mangrulkar et al., 2022) that can handle multiple sequentially introduced marketplaces, outperform marketplace-specific training, and significantly reduce computational resources and training time. We also aim to enable effective knowledge transfer across different marketplaces. The paper is organized as follows: Section 4.1 discusses using language-specific adapters and fusion modules in a continual learning environment. Section 4.2 introduces our proposed architecture, Attention Distillation, which distills attention scores generated from fusion layer with the previously trained fusion layer to prevent catastrophic forgetting and enhance performance. Section 4.3 explores how translation enhancement improves performance.

4.1 Adapters and Adapter Fusion Modules in CL Context

Adapters: The base model (Θ_{base}) is a transformer-based, multi-language pre-trained architecture (e.g., mBERT) with all parameters frozen. When a new language is introduced, a randomly initialized adapter, based on the Pfeiffer architecture (Pfeiffer et al., 2020b), is added after the feed-forward layer in each mBERT layer (see Figure 2(a)). A classification head is placed on the final adapter layer, and the new adapter is trained on marketplace data (D_t). Once training is complete, the adapter is preserved independently, with its weights denoted as Θ_{A_t}, where t corresponds to the time-step. The model weights are expressed as:

$$\Theta_{M_t} = \Theta_{base} + \sum_{j=1}^{t} \Theta_{A_j} \quad (5)$$

During training, only the adapter weights (Θ_{A_t}) are unfrozen, while all other parameters remain frozen. The training objective for model M_t is as follows:

$$\Theta_{A_t} \leftarrow \underset{\Theta}{\text{argmin}} \mathcal{L}_t(D_t; \Theta_{base}, \Theta_{A_1}, ..., \Theta_{A_{t-1}}, \Theta) \quad (6)$$

Adapter Fusion: To enable knowledge sharing between different language adapters, an attention layer called adapter fusion is added on top of the adapters (Pfeiffer et al., 2020a) (see Figure 2(a)). Let Ψ_t denote the Key, Value, and Query matrices introduced by the fusion layer upon the introduction of the D_t marketplace. After training an adapter on D_t, the entire model, including adapters, is frozen. The Adapter Fusion layer is then trained with task-specific loss $\mathcal{L}_t$, and the learning objective becomes:

$$\Psi_t \leftarrow \underset{\Psi}{\text{argmin}} \mathcal{L}_t(D_t; \Theta_{base}, \Theta_{A_1}, ..., \Theta_{A_t}, \Psi) \quad (7)$$

The final model weights are:

$$\Theta_{M_t} = \Theta_{base} + \sum_{j=1}^{t} \Theta_{A_j} + \Psi_t \quad (8)$$

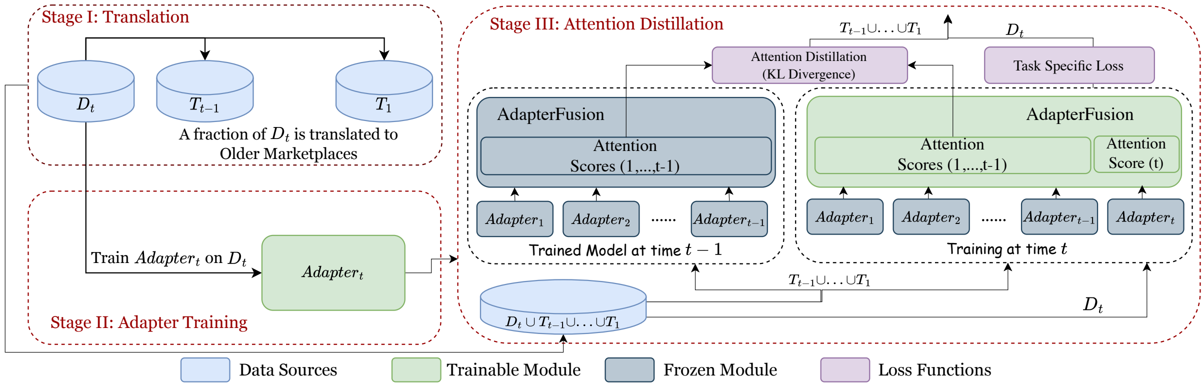
Stage I: Translation
D_t
T_{t-1}
T_1
A fraction of D_t is translated to Older Marketplaces
Train $Adapter_t$ on D_t
$Adapter_t$
Stage II: Adapter Training
Stage III: Attention Distillation
$T_{t-1} \cup \ldots \cup T_1$
D_t
Attention Distillation (KL Divergence)
Task Specific Loss
AdapterFusion
Attention Scores (1,...,t-1)
$Adapter_1$
$Adapter_2$
......
$Adapter_{t-1}$
Trained Model at time $t-1$
AdapterFusion
Attention Scores (1,...,t-1)
Attention Score (t)
$Adapter_1$
$Adapter_2$
......
$Adapter_{t-1}$
$Adapter_t$
Training at time t
$T_{t-1} \cup \ldots \cup T_1$
$D_t \cup T_{t-1} \cup \ldots \cup T_1$
D_t
Data Sources
Trainable Module
Frozen Module
Loss Functions

Figure 3: Three-Stage Training Pipeline for the Model (M_t) at Time t within a Continual Learning Framework: Translation followed by individual Adapter training followed by Attention Distillation Method.

4.2 Attention Distillation Training using Adapter Fusion layer

The fusion layer's weight dimension changes with the addition of new language-specific adapters, making it impossible to reuse trained weights from model M_{t-1} in the new model M_t. To address this, our attention distillation method follows these steps:

1. When a new language is introduced at time t, the fusion layer parameters are randomly initialized. The previous model (M_{t-1}) serves as the teacher, and the new model (M_t) as the student.

2. During training, both models process each batch of data. The student model's attention scores ($s_{l,j}$ in Equation 1) for old adapters are distilled from the teacher model using KL-Divergence (Kullback and Leibler, 1951), while scores for the new adapter are trained using the conventional approach (outlined in (Vaswani et al., 2017)) for their respective new target tasks.

Let Ω_t represent the attention score matrix produced by the Fusion layer in M_t with dimensions $batch\_size \times max\_tokens \times t$. The attention distillation loss ($\mathcal{L}_{AD}$) and total loss ($\mathcal{L}_{total\_t}$) are defined as follows, where ω_t = $\{\Omega_t[i,j,k]; i \leq batch\_size, j \leq max\_tokens,$ $k \leq t\text{ - }1\}$.

$$\mathcal{L}_{AD} = KL(\Omega_{t-1}||\omega_t) = \sum_{i,j,k} \Omega_{t-1} \log(\frac{\Omega_{t-1}}{\omega_t}) \quad (9)$$

$$\mathcal{L}_{total\_t} = \mathcal{L}_t + \mathcal{L}_{AD} \quad (10)$$

The total loss balances task-specific learning for the new marketplace with maintaining attention distribution from the previous model.

4.3 Proposed Method: Attention Distillation with translation

When training model M_t with a new language, we compute the attention distillation loss using model M_{t-1} (teacher) and M_t (student). However, for syntactically different languages, the activations from M_{t-1}'s fusion layer may become irrelevant. To address this, our revised method includes the following steps:

1. Train a new language-specific adapter (Θ_{A_t}) using dataset D_t, incorporating it into M_t, which already includes the frozen base model (Θ_{base}) and $t-1$ language-specific adapters.

2. Translate a portion of D_t into older languages, denoted as $T_1, T_2, ..., T_{t-1}$.

3. Introduce a fusion layer (Ψ_t) atop the t adapters in M_t, freezing all parameters except the fusion layer.

4. During training, if a batch is from the translated subset, we pass it to both the teacher (M_{t-1}) and student (M_t) models, applying the attention distillation loss. If the batch is from the new language D_t, we compute the task-specific cross-entropy loss.

To manage computational complexity, only a small subset of the new data is translated into older languages. The learning objective is updated as:

$$\Psi_t \leftarrow \underset{\Psi}{\text{argmin}}\{\mathcal{L}_t(D_t, \Psi) + \mathcal{L}_{AD}(T_1, T_2, ..., T_{t-1}, \Psi)\} \quad (11)$$

Algorithm 1 in Appendix and Figure 3 provide an overview and depict our proposed approach.

5 EMPIRICAL EVALUATION

We present our findings on the benefits of using multi-lingual continual learning for relevance classification tasks. We begin with the dataset details.

Method	ROC-AUC				#Trainable Parameters
	M_A	M_B	M_C	M_D	
SM (baseline)	0.880(±0.0009)	0.8540(±0.0001)	0.8712(±0.0006)	0.8760(±0.0002)	110M
Sequential Fine-tuning					
$M_A \rightarrow M_B$	0.8756(±0.0007)	0.8630(±0.0003)	–	–	
$M_A \rightarrow M_B \rightarrow M_C$	0.8670(±0.0001)	0.850(±0.0011)	0.8851(±0.0004)	–	
$M_A \rightarrow M_B \rightarrow M_C \rightarrow M_D$	0.8583(±0.0010)	0.8429(±0.0006)	0.8701(±0.0012)	0.8824(±0.0005)	110M
Adapter and Adapter Fusion - without Sequential					
Adapter	0.8742(±0.0003)	0.8532(±0.0009)	0.8693(±0.0008)	0.870(±0.0013)	0.59M
Adapter Fusion	0.8867(±0.0002)	0.8756(±0.0006)	0.8832(±0.0005)	0.8851(±0.0008)	22M
*Attention Distillation with Translation (**Our approach**)*					
$M_A \rightarrow M_B$	0.8829(±0.0004)	0.8782(±0.0003)	–	–	
$M_A \rightarrow M_B \rightarrow M_C$	0.8832(±0.0007)	0.8724(±0.0002)	0.8890(±0.0005)	–	
$M_A \rightarrow M_B \rightarrow M_C \rightarrow M_D$	0.8874(±0.0012)	0.8768(±0.0009)	0.8854(±0.0010)	0.8865(±0.0004)	22M

Table 1: ROC-AUC scores for SM, sequential fine-tuning, adapters and adapter fusion (not in sequence), and our proposed method on the Amazon proprietary Dataset. We also include the number of trainable parameters for each method. The sequence x → y → z indicates the fine-tuning order of the mBERT model, where after training on the z language, performance is evaluated on all languages, x, y, and z. Green signifies a ROC-AUC score increase compared to the SM baseline, while red indicates a decrease. Mean & std. (±) error for ROC-AUCs are reported based on 5 trials.

Method	Amazon Proprietary Dataset				Aicrowd Public Dataset		
	M_A	M_B	M_C	M_D	En	Es	Jp
HAT	0.8349(±0.0005)	0.8367(±0.0012)	0.8438(±0.0010)	0.8427(±0.0008)	0.7768(±0.0002)	0.7271(±0.0002)	0.7242(±0.0001)
CTR	0.8538(±0.0011)	0.8221(±0.0008)	0.8338(±0.0009)	0.8346(±0.0004)	0.7855(±0.0013)	0.7400(±0.0011)	0.7258(±0.0003)
B-CL	0.8421(±0.0002)	0.8349(±0.0002)	0.8389(±0.0004)	0.8410(±0.0007)	0.7623(±0.0006)	0.7382(±0.0004)	0.7244(±0.0008)
DyTox	0.8740(±0.0002)	0.8642(±0.0004)	0.8702(±0.0005)	0.8654(±0.0006)	0.7624(±0.0010)	0.7483(±0.0003)	0.7168(±0.0007)
Attention Distillation	**0.8852**(±0.0008)	**0.8727**(±0.0001)	**0.8738**(±0.0008)	**0.8772**(±0.0002)	**0.8004**(±0.0001)	**0.7894**(±0.0012)	**0.7400**(±0.0013)

Table 2: Comparing ROC-AUC with SOTA Continual Learning Models on both the Amazon proprietary dataset and a publicly available Aicrowd query dataset. **The ROC-AUC values are averaged over 4 random sequences**. Mean & std. (±) error for ROC-AUCs are reported based on 5 trial runs.

Datasets: 1. Amazon proprietary e-commerce data from four marketplaces: To ensure confidentiality, we denote the four marketplaces as M_A, M_B, M_C, and M_D. Each marketplace dataset includes a ground truth label categorized as either relevant or non-relevant. All datasets in our analysis are anonymized, aggregated, and do not represent production distribution. **2.** Public Aicrowd Shopping Query dataset (Reddy et al., 2022) from EN, ES, and JP marketplaces. Further details on the generation of these datasets can be found in Appendix A.

Reproducibility and Hyperparameters: Please refer to Appendix B for detailed information on the reproducibility of our experiments and the hyperparameter configurations.

Algorithm Baselines: To evaluate our method, we use the following baselines:

(i) Single Marketplace (SM): Fine-tuning M-BERT individually for each marketplace dataset.

(ii) Sequential Fine-tuning: Sequentially fine-tuning M-BERT for each marketplace in a specific order.

(iii) HAT (Serra et al., 2018): A hard attention mechanism that retains previous tasks' information while learning new tasks.

(iv) CTR (Ke et al., 2021a): Incorporates a continual learning plug-in (CL-plugin) in BERT to facilitate knowledge transfer and protect task-specific knowledge.

(v) B-CL (Ke et al., 2021b): Uses continual learning adapters and capsule networks to promote knowledge transfer and safeguard task-specific knowledge.

(vi) DyTox (Douillard et al., 2022): A dynamic continual learning strategy with a transformer-based architecture.

Evaluation Metric: For classifying relevance and identifying optimal query-product pairs, we use ROC-AUC (Brown and Davis, 2006) as our primary metric. Although ranking metrics like precision@k, recall@k, and NDCG could be used, however, we opted not to generate results for ranking metrics due to the limited number of products per query in our datasets.

5.1 Results

In Table 1, we present our proposed method results on Amazon proprietary dataset, comparing them with- SM, Sequential Fine-tuning, and Adapter & Adapter Fusion (non-sequential). Throughout our experiments, we use the pre-trained mBERT model. Sequential Fine-tuning demonstrates a case of catastrophic forgetting for all the older marketplaces. **Regarding RQ1**, Adapter fusion trained on all marketplaces together demonstrates superior results compared to SM with an ∼80% reduction in parameters. However, it cannot be employed in a Continual fashion. Conversely, our proposed method, specifically tailored for Continual fine-tuning, surpasses the SM baseline and achieves nearly comparable performance with Adapter fusion while reducing parameters by ∼80%.

RQ2: Comparison with SOTA CL methods: Table 2 highlights that the current SOTA continual learning models are not well equipped for handling multilingual continual learning scenarios. This can be attributed to the architecture of some methods such as CTR (Ke et al., 2021a) which weighs the embeddings generated by the base transformer model based on the similarity between different tasks. Since the task remains the same, the respective capsules in CTR are unable to capture any additional information that needs to be transferred between different marketplaces and hence we notice that the results are similar for every marketplace even though the data distribution is significantly different. In contrast, our method consistently outperforms all SOTA continual learning methods when provided with a multilingual continual learning scenario.

RQ3: Benefits with Translation: Translating the entirety of the new marketplace's data back into the old marketplace languages significantly extends the time required for training. In this context, we present a summary of our findings in Figure 4. We employ our proposed approach for a sequence of four languages, translating data from the fourth language into the first three. We then calculate the average ROC-AUC gains for the initial three languages, taking into account the percentage of data translated. The findings reveal that the highest performance coupled with the ideal training duration is attained when 30% of the entire dataset is translated back into the older marketplace languages.

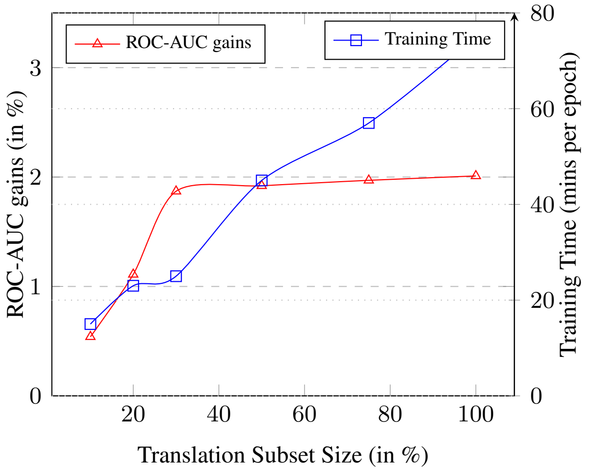

Figure 4: Training Time and ROC-AUC Gains vs. Translation Subset Size

5.2 Deployment Considerations

Effective query-product relevance models are crucial for reducing irrelevance on online e-commerce stores. Our analysis shows that a significant portion of product impressions come from offline sourcing strategies, which contribute substantially to search irrelevance. We use various offline strategies to curate product lists for head and torso queries, which are repetitive and cover a large portion of query coverage. We then apply a high-performing relevance model to evaluate query-product pairs, storing highly relevant pairs in an offline cache. This relevance model enhances the relevance of displayed query-product pairs, leading to improved customer experience and an increase in overall sales.

6 Conclusion and Future Work

We propose a novel Attention Distillation method and outline a training process for multilingual continual learning. This method enables the seamless integration of new marketplaces over time without causing a decline in performance for older ones. Our experiments on internal and external datasets demonstrate consistent performance across all marketplaces, outperforming state-of-the-art Continual Learning methods. This approach also offers potential for future exploration in applying Attention Distillation to multi-task problem-solving challenges.

References

Sanjay Agrawal, Srujana Merugu, and Vivek Sembium. 2023a. Enhancing e-commerce product search through reinforcement learning-powered query reformulation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4488–4494.

Sanjay Agrawal, Vivek Sembium, and MS Ankith. 2023b. Kd-boost: Boosting real-time semantic matching in e-commerce with knowledge distillation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 131–141.

Christopher Brown and Herbert Davis. 2006. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80:24–38.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Preprint*, arXiv:1906.01076.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. 2022. Dytox: Transformers for continual learning with dynamic token expansion. *Preprint*, arXiv:2111.11326.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021a. Achieving forgetting prevention and knowledge transfer in continual learning. *Preprint*, arXiv:2112.02706.

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

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *Preprint*, arXiv:1606.09282.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2022. Gradient episodic memory for continual learning. *Preprint*, arXiv:1706.08840.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. *Preprint*, arXiv:1711.05769.

Sourab Mangrulkar, Ankith M S, and Vivek Sembium. 2022. Multilingual semantic sourcing using product images for cross-lingual alignment. In *The Web Conference 2022*.

Michinari Momma, Chaosheng Dong, and Yetian Chen. 2022. Multi-objective ranking with directions of preferences.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *ArXiv*, abs/2005.00247.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *Preprint*, arXiv:2005.00052.

Amal Rannen, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017a. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017b. icarl: Incremental classifier and representation learning. *Preprint*, arXiv:1611.07725.

Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale ESCI benchmark for improving product search.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference. *Preprint*, arXiv:1810.11910.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2022. Progressive neural networks. *Preprint*, arXiv:1606.04671.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557. PMLR.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. *Preprint*, arXiv:2112.08654.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

A Datasets

1. Amazon Proprietary Dataset From four distinct Amazon marketplaces, we gather separate sets of 500K human-audited query-product pairs, each containing data in its respective language. Test and validation datasets are generated by randomly sampling 30K query-product pairs for each marketplace, and these 60K pairs are then excluded from the training dataset.

2. Aicrowd Shopping Query Public Dataset (Reddy et al., 2022) is a publicly available dataset released by Amazon containing product search data for the EN, ES and JP marketplaces. To create test and validation datasets, 20% of the training datasets are chosen at random and excluded from the training datasets. Each query-product pair is annotated with labels denoted as E/S/C/I, which stand for Exact, Substitute, Complement, and Irrelevant. In the context of search, the pairs labeled as Exact and Substitute are considered relevant (positive class), while the pairs labeled as Complement and Irrelevant are considered irrelevant (negative class). Hence, the task can be formulated as a binary classification problem, with the goal of comparing performance in terms of roc-auc.

B Reproducibility and Hyperparameters

In this section, we present the hyperparameters and training methodologies used in our experiments. We use publicly available datasets and open-source models to ensure that our work can be independently verified and reproduced. All experiments are carried out utilizing the PyTorch framework (Paszke et al., 2019) in conjunction with the HuggingFace models (Wolf et al., 2019). We use a consistent set of hyperparameters during training on Proprietary and Public datasets, which were optimized through a series of preliminary trials and are detailed in Table 3.

The bert-base-multilingual-uncased (Devlin et al., 2018) [1] model serves as the base model for conducting all the CL-based experiments. During the training phase, we employ pre-trained checkpoints and then train every marketplace adapter for 5 epochs followed by training the Adapter Fusion layer using Attention Distillation for an additional 5 epochs, incorporating early-stopping criteria. Regarding the translation-based distillation process detailed in Section 4.3, when addressing a new language, we execute translation on 30% of the data to revert it back into the languages of earlier marketplaces. This is accomplished using Helsinki-NLP's Opus MT models (Tiedemann and Thottingal, 2020). Please note that our methodology demands significantly less computational resources as compared to the baseline models since the weights of the base transformer model are frozen in our training process.

Hyperparameter	**Value**
Batch Size	512
Learning Rate	5e-5
Epochs for Adapter Training	5
Epochs for Adapter Fusion Training	5
Weight Decay	0.0
Optimizer	Adam
Adam ϵ	1e-8
Gradient Clipping	0.1

Table 3: Hyperparameters used for training the models.

[1] https://huggingface.co/bert-base-multilingual-uncased

Algorithm 1 Training Procedure for the Model M_t Using Attention Distillation with Translation Approach in a Continual Learning Context

```
Require: Dataset $D_t$, Translated Datasets $\{T_1, ..., T_{t-1}\}$, Adapter $A_t$, Batchsize $bs$, Task Specific Loss
    $\mathcal{L}_t$, Max Token Length $v$, KL Divergence Loss $KL$, Frozen Base Model Parameters $\Theta_{base}$
Ensure: Learn Adapter Parameters $\Theta_{A_t}$ and Adapter Fusion Parameters $\Psi_t$ at time-step $t$
    $\Theta_{A_1}, ..., \Theta_{A_{t-1}} \leftarrow$ Frozen Adapters Parameters
    $\Theta_{A_t} \leftarrow \underset{\Theta}{\text{argmin}}\, \mathcal{L}_t(D_t, \Theta_{base}, \Theta_{A_1}, ..., \Theta_{A_{t-1}}, \Theta)$
    $\Theta_{A_t} \leftarrow$ Frozen $t^{th}$ Adapters Parameters
    $\Psi_{t-1} \leftarrow$ Frozen Adapter Fusion Parameters
    $\Omega_t \leftarrow$ Attention score matrix from Adapter Fusion
    $\{Q_{bs=1}, ..., Q_{bs=last}\} \in D_t \bigcup T_1 \bigcup ... \bigcup T_{t-1}$
    for j $\leftarrow$ 1 to bs=last do
        $\omega_t \leftarrow \{\Omega_t[p, q, r]; p \leq bs, q \leq v, r \leq \textit{t - 1}\}$
        $\mathcal{L}_{AD} \leftarrow KL(\Omega_{t-1}||\omega_t)$
        $\Psi_t \leftarrow \underset{\Psi}{\text{argmin}}\{\mathcal{L}_t(D_t, \Psi) + \mathcal{L}_{AD}(T_1, ..., T_{t-1}, \Psi)\}$
    end for
```