Informatics Project Proposal 2024

Caching Large Model Requests with Dynamic Request Streams using Continual Learning

Shriram Piramanayagam

This research project explores the optimization of neural caching using Continual Learning (CL) techniques to enhance online accuracy and reduce the costs associated with Large Language Model (LLM) queries. By integrating Active Learning and Knowledge Distillation, the project aims to effectively handle dynamic request streams without compromising data privacy or incurring high computational costs. The outcomes are expected to significantly advance the efficiency of neural caching in real-world applications, potentially reducing reliance on expensive LLM interactions.

Supervisor
Prof Ivan Titov

IPP Tutor
Shangmin Guo

THE UNIVERSITY *of* EDINBURGH
informatics

Table of contents

List of figures

List of tables

1 Introduction

1.1 Motivation

Large Language Models (LLMs) offer advanced natural language understanding abilities. LLMs are pre-trained on a large corpus of text data. During this phase, they learn diverse language patterns and structures through unsupervised language modelling, where the model predicts subsequent words based on the preceding ones. After pre-training, the model is fine-tuned for specific tasks like answering questions or generating text. However, the cost associated with training and managing LLMs is substantial, rendering it prohibitive for most organizations. LLMs excel in in-context learning, making them highly effective in zero-shot and few-shot scenarios. This capability is valuable for applications where fine-tuning a model for specific tasks is not viable due to the limited availability of labelled data or the need for rapid solutions to emerging challenges. For example, in the context of on-device learning, it's not feasible for the user to label a large volume of data, necessitating the need for few-shot approaches.

Compared to Large Language Models (LLMs), smaller models often require additional fine-tuning with task-specific labelled data to perform effectively. In scenarios where labelled data is unavailable, organizations subscribe to LLM providers. These subscriptions enable them to utilize LLMs through APIs, incurring charges on a per-call basis. For example, customer support chatbots use LLMs to discern user intentions and provide relevant assistance (Ham et al., 2020). Over time, the prompts given to the LLM may become repetitive. Also, it exposes the entire request stream to the LLM provider. To address issues of cost and data privacy, a new framework has been studied where a smaller, locally run model known as the *student* model is periodically trained using the outputs from the LLM, or *teacher* model (Ramírez et al., 2023; Stogiannidis et al., 2023). This method, also known as *neural caching*, enables the student model to increasingly handle requests independently, significantly cutting costs while only slightly reducing performance (Stogiannidis et al., 2023).

Existing neural caching frameworks were only evaluated with static request streams, assuming a constant and unchanging probabilistic distribution of requests over time. This assumption is far from realistic in real-world applications where data streams may not be static or independent and identically distributed (iid). We hypothesize that the neural caching with a dynamic request stream is a **continual learning** (CL) problem. CL focuses on incrementally training models on a stream of data to accumulate knowledge over time. Consequently, we aim to study the efficacy of Active Learning policies in the context of dynamic request streams using Continual Learning.

1.2 Literature review

Continual Learning: CL allows machine learning models to learn continuously on new data and enables learning from a stream of data. The primary challenge it addresses is catastrophic forgetting (McCloskey and Cohen, 1989) which is the tendency of a neural network to forget old knowledge when new learning occurs. There are two naive approaches to solving the continual learning problem (Verwimp et al., 2024). The first one involves incrementally training or fine-tuning a model exclusively on newly acquired data. This method often leads to less than optimal performance, as models may become overly adapted to recent data at the expense of previously learned information. The second approach entails periodically retraining the model on the entire dataset accumulated up to that point. While this method may preserve earlier learning better, it is undesirable due to its high computational and memory costs. Continual learning aims to develop methodologies that achieve a better trade-off between performance and resource efficiency, such as computational and memory usage, compared to these naive approaches.

According to Harun et al. (2023), the majority of continual learning techniques are not specifically designed to maximize computational efficiency. However, they demonstrate that it is feasible to significantly decrease training times by an order of magnitude while achieving comparable levels of performance. Successful implementation of continual learning methodologies could substantially lower both the financial costs and the considerable carbon emissions that are typically associated with retraining models from scratch (Amodei and Hernandez, 2023) without compromising on accuracy.

Neural caching: The implementation of neural caching by Stogiannidis et al. (2023) incorporates a kNN classifier as the *student* model. This model choice is notable for its operational simplicity as it does not require re-training. However, as the dataset expands, the kNN classifier's reliance on accessing the entire dataset for each decision introduces significant memory challenges, emphasizing the trade-off between model simplicity and scalability. Ramírez et al. (2023) evaluates various active learning-based selection criteria for the policy, with Margin Sampling and Query by Committee showing consistent benefits across tasks and budgets. In situations where data undergoes significant changes over time, or in cases where accuracy takes precedence over real-time performance, such as financial models or medical diagnostic systems, complete re-training might be the preferred approach. Here, neural caching setup is an online form of Active Learning with Knowledge Distillation. Both Active Learning (Settles, 2009) and Knowledge distillation (Bucilua, Caruana and Niculescu-Mizil, 2006; Hinton, Vinyals and Dean, 2015) focus on optimising the final accuracy of the student model, whereas the neural caching problem aims to optimise the online accuracy of the entire setup.

2 Methodology

The objective of neural caching is to enhance the online accuracy of the system while efficiently training the student model within a fixed budget allocated for utilizing a Large Language Model (LLM). This study will focus on the incorporation of Continual Learning (CL) techniques that enhance the model's adaptability to new data and its resilience against forgetting. Such capabilities are particularly vital in dynamic environments where both accuracy and efficiency are paramount.

Neural caching aims to generate class labels for a sequence of inputs $(x_1, \ldots, x_n)$, from the input space X. The student model undergoes updates through Continual Learning (CL) methods every f processed request, where f denotes the frequency of model updates. The proposed system will simulate a dynamic stream of requests, processing each in real-time either through the student model or the LLM, depending on the established policy. During the learning phase, access to ground truth will not be assumed with the aim to mimic a fully automated online scenario similar to Ramírez et al. (2023). This setup will allow for the exploration of the trade-offs between immediate performance and long-term learning across a range of operational contexts. The entire process is shown in Figure Appendix B and the algorithm is described in Appendix A.

2.1 Instance Selection Criteria

The experiment aims to study the efficiency of Active Learning (AL) policies in the context of dynamic request streams using Continual Learning (CL). Ramírez et al. (2023) show that Margin Sampling and Query by Committee perform well in the neural caching setup, among the AL strategies. For simplicity, we will use Margin Sampling in this study in addition to Front Loading and Random baselines.

Front-loading (FR) involves utilizing the entire annotation budget initially by labelling all instances through the Large Language Model (LLM). Once the budget is depleted, subsequent queries are managed exclusively by the student model.

Margin Sampling (MS) (Luo et al., 2005; Scheffer, Decomain and Wrobel, 2001) focuses on selecting examples where the difference between the probabilities of the top two predicted labels by the student model is the largest. The margin is calculated as:

$$\mathrm{Margin}(x_i) = \log P(y_i = k_1^*|x_i) - \log P(y_i = k_2^*|x_i) \tag{1}$$

where k_1^* and k_2^* represent the labels with the highest and second-highest likelihoods, respectively, as determined by the probability distribution $P(y_i|x_i)$ provided by the student model. Recent evaluations by Schröder, Niekler and Potthast (2021) have confirmed the efficacy of MS, particularly highlighting its superior performance with Transformer models (Devlin et al., 2019) in an offline, pool-based setting. To apply MS, as well as other criteria, in an online environment as a selection policy, a threshold is established. Only examples with a margin exceeding this threshold are selected until the budget is depleted.

2.2 Datasets

We will follow (Ramírez et al., 2023) and use the following datasets: ISEAR (Shao, Doucet and Caruso, 2015); RT-Polarity (Pang and Lee, 2005); fact-checking dataset FEVER (Thorne et al., 2018); and the question-answering dataset Openbook (Mihaylov et al., 2018). All datasets are divided into online and test sets with an 80%-20% split, except for Openbook due to its smaller size. The classes are uniformly distributed across all the datasets.

ISEAR (Shao, Doucet and Caruso, 2015) contains data collected over many years during the 1990s by a large group of psychologists worldwide. The dataset includes reports on seven major emotions by close to 3000 respondents in 37 countries on all 5 continents (classes: joy, fear, shame, sadness, guilt, disgust, anger; 7666 examples)

RT-Polarity (Pang and Lee, 2005) is a collection of movie-review documents labelled for their overall sentiment polarity (classes: positive, negative; 10662 examples).

FEVER (Thorne et al., 2018) is a fact-checking dataset with claims that can be checked from Wikipedia (classes: true, false; 6612 examples)

Openbook (Mihaylov et al., 2018) is a demanding question-answering dataset, designed to evaluate human comprehension of a topic, similar to open book exams. Each entry is composed of a multiple-choice question (classes: A, B, C, D) and contains one fact that can assist in answering the question. The entire dataset comprises 5957 data points, with 5457 chosen for the online set and 500 reserved for testing.

Class incremental datasets: We will re-sample the ISEAR, RT-Polarity and FEVER datasets in a class incremental way with samples from a single class grouped. Openbook is not chosen for this setup since its classes have no inherent meaning.

Task incremental datasets: A key requirement for Continual Learning (CL) is its ability to handle various task sequences robustly, given that the order of tasks is not predetermined. Therefore, we will conduct all our experiments with the online portion of the data using a variety of task sequences and present the average results similar to (Mehta et al., 2021).
Seq 1. RT-Polarity->Openbook->FEVER->ISEAR
Seq 2. ISEAR->FEVER->Openbook->RT-Polarity
Seq 3. Openbook->RT-Polarity->ISEAR->FEVER

The accuracy of LLM on these datasets as per Ramírez et al. (2023) can be approximated as a simple difficulty metric for the task and the first and second sequences are based on the increasing and decreasing order of this difficulty metric respectively. The third sequence is random.

2.3 Continual Learning Methods

We will use Fine-tuning and Elastic Weight Consolidation as the Continual Learning (CL) methods for our study.

Fine-tuning (FT): The model will be sequentially fine-tuned using LoRA adapters after every f processed request on each task. Fine-tuning is susceptible to the *catastrophic forgetting* (McCloskey and Cohen, 1989) problem due to the absence of additional learning constraints and will be considered as the baseline for our experiments.

Elastic Weight Consolidation (EWC): (Kirkpatrick et al., 2017) EWC allows neural networks to retain old knowledge by slowing down learning on weights important for previous tasks. The importance of weights is determined using the Fisher Information Matrix. EWC is shown to enable continual learning in neural networks, allowing them to learn sequential tasks without forgetting earlier ones.

2.4 Experiment Details

An experimental setup similar to Ramírez et al. (2023) will be followed. For consistency, the frequency of updating the student model, designated by f, will be set to 1000, and each query will incur a uniform cost, denoted by $c(x_i) = 1$. To mitigate the cold start problem, the initial model, S_0, will be pre-trained using data points derived from the Large Language Model (LLM) with $N = 100$ for the ISEAR and RT-Polarity datasets, and $N = 1000$ for FEVER and Openbook, so that S_0 performs above the baseline of random selection.

$T5_{base}$ (Raffel et al., 2019) will be used as the backbone for the student model. We will freeze the model weights and add LoRA adapter layers, which will allow for parameter-efficient fine-tuning without altering the underlying model weights (Hu et al., 2022). The budget range for a task (b_{task}) will be varied from 1000 to 3500 in increments of 500. We will use the code[1] and LLM annotated dataset[2] released by Ramírez et al. (2023) as a starting point for the experiments along with the hyperparameters.

[1] https://github.com/guillemram97/neural-caching

[2] https://huggingface.co/datasets/guillemram97/cache_llm

2.5 Expected Outcomes

We hypothesize that neural caching with dynamic request streams is a Continual Learning (CL) problem. The experiments are expected to confirm this hypothesis where the CL methods such as Elastic Weight Consolidation (EWC) should alleviate the forgetting problem and provide better average accuracy (A) and less forgetting measure (F) when compared to plain fine-tuning in both class incremental and especially in task incremental scenarios. This would ultimately confirm the performance benefits and robustness of CL in the neural caching framework for dynamic data streams.

2.6 Evaluation

As per Ramírez et al. (2023), average accuracy will be calculated for class incremental datasets by dividing the Area Under The Curve (AUC) by the budget range. Moreover, the accuracy of the final student model will be assessed on a separate test dataset to measure its *final accuracy*. For task incremental experiments, we will follow Lopez-Paz and Ranzato (2017) to evaluate our setup using average accuracy and forgetting measure which evaluates the forward and backward transfer of knowledge in our setup respectively.

Average Accuracy (A): The average accuracy after incremental training from the first task to T is given as:

$$A_T = \frac{1}{T}\sum_{i=1}^{T} a_{T,i} \tag{2}$$

where $a_{T,i}$ is the performance of the model on the test set of task t_i.

Forgetting Measure (F): The average forgetting measure after incremental training from the first task to T is defined as:

$$F_T = \frac{1}{T-1}\sum_{i=1}^{T-1} f_{T,i} \tag{3}$$

where $f_{T,i}$ is the forgetting on task t_i after the model is trained up to task t_j and computed as:

$$f_{j,i} = \max_{k\in\{1,\dots,j-1\}} a_{k,i} - a_{j,i} \tag{4}$$

3 Work plan

The project will be managed using an iterative approach, with mandatory weekly meetings every Thursday with the Supervisor to assess progress against milestones and adapt plans as necessary. Access to required code, datasets and computer resources has already been acquired.

3.1 Plan, milestones and deliverables

The project management plan delineated in Figure 1 systematically organizes the progression of the project through key milestones. The initial phase involves the completion of background work by Week 2 (M1), setting the groundwork for the project. By Week 5 (M2), the coding is expected to be complete, and feedback from the second marker is anticipated, resulting in completed experiment code. The mid-project phase aims for the completion of experiments by Week 7 (M3), transitioning into the evaluation phase by Week 9 (M4), culminating in a draft of the report. The final phase targets Week 10 (M5) for the submission of the final report. The deliverables for each phase are outlined in Table 2.

Risk Name	Severity	Likelihood	Mitigation
Unable to implement CL algorithms	Medium	Medium	Use alternative algorithms
Pipeline not working as expected	High	Medium	Extend implementation phase
Experiments give negative results	Medium	Low	Research the possible reasons

Table 1 Risk assessment table..

3.2 Risk assessment

The risk assessment plan shown in Table 1 identifies potential impediments to the project's successful completion. By Week 4 (R1), there is a risk identified concerning the implementation of the Continual Learning (CL) algorithm, which could hinder progress. The operational functionality of the pipeline is a concern by Week 6 (R2), and there is a contingency for the possibility of experiments yielding negative results by Week 9 (R3). These risks are assessed for their probability and potential impact on the project, and preemptive measures are taken to mitigate them, ensuring that project objectives are met within the stipulated timeline.

4 Responsible Research

The experiment under consideration does not introduce novel algorithms or datasets, and as such, the immediate societal and ethical implications are minimal. Nonetheless, the broader implications of continual learning methodologies, which this paper scrutinizes, could be substantial over time. For instance, the reduction of computational costs in machine learning, as posited in the study, has the potential to diminish the environmental footprint of this technology. Continual Learning has the promise of making machine learning model training more accessible, thus fostering a more inclusive landscape for technological advancement. On the flip side, the same flexibility that allows for rapid updates to network segments could be co-opted by nefarious entities to deliberately disseminate misinformation, which could result in misinformed public opinions or even hazardous outcomes. The starter data and code described in 2.4 to be used in the project are license-free. Finally, we will release the modified code and data created for this study through GitHub to encourage further research on this problem.

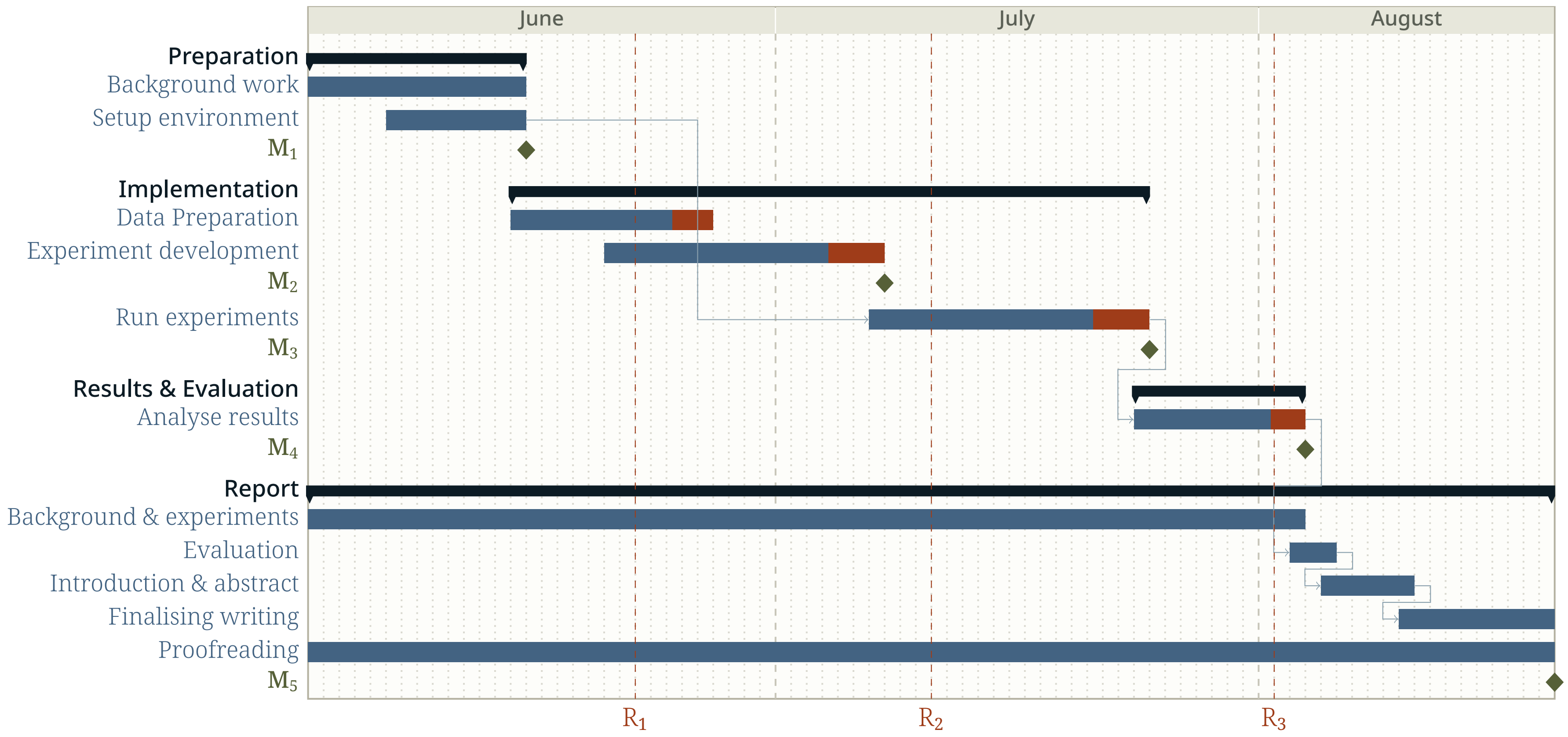

Figure 1 Gantt Chart of project activities, see Table 2 for details. Overarching bars coloured ■ represent the maximum time (including contingency) allocated per phase, 🟦 represents the scheduled time for a task and 🟥 its contingency time. Arrowed lines indicate task dependencies.

	Event	Week(s)	Description	Deliverable
Milestone	M_1	2	Background work completed	
	M_2	**5**	**Complete coding and Second marker feedback**	Experiment code
	M_3	7	Experiments completed	
	M_4	9	Evaluation completed	**Report draft**
	M_5	**10**	**Submission deadline**	**Final report**
Risk	R_1	4	Unable to implement CL algorithm	
	R_2	6	Pipeline not working as expected	
	R_3	**9**	**Experiments give negative results**	

Table 2 Project milestones, deliverables, and risks (**bold** = main); see Figure 1.

5 References

Amodei, Dario and Hernandez, Danny, 2023. **AI and compute** [Online]. Available from: https://openai.com/research/ai-and-compute [Accessed 21 April 2024] (cited on page 2).

Bucilua, Cristian, Caruana, Rich and Niculescu-Mizil, Alexandru, 2006. Model compression. **Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining** [Online], Kdd '06. Philadelphia, PA, USA: Association for Computing Machinery, pp.535–541. Available from: https://doi.org/10.1145/1150402.1150464 (cited on page 3).

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton and Toutanova, Kristina, 2019. Bert: pre-training of deep bidirectional transformers for language understanding. arxiv. **Arxiv preprint arxiv:1810.04805** (cited on page 4).

Ham, Donghoon, Lee, Jeong-Gwan, Jang, Youngsoo and Kim, Kee-Eung, 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. **Proceedings of the 58th annual meeting of the association for computational linguistics**, pp.583–592 (cited on page 1).

Harun, Md Yousuf, Gallardo, Jhair, Hayes, Tyler L., Kemker, Ronald and Kanan, Christopher, 2023. **Siesta: efficient online continual learning with sleep**. arXiv: 2303.10725 [cs.CV] (cited on page 2).

Hinton, Geoffrey, Vinyals, Oriol and Dean, Jeff, 2015. Distilling the knowledge in a neural network. **Arxiv preprint arxiv:1503.02531** (cited on page 3).

Hu, Edward J, shen, yelong, Wallis, Phillip, Allen-Zhu, Zeyuan, Li, Yuanzhi, Wang, Shean, Wang, Lu and Chen, Weizhu, 2022. LoRA: low-rank adaptation of large language models. **International conference on learning representations** [Online]. Available from: https://openreview.net/forum?id=nZeVKeeFYf9 (cited on page 6).

Kirkpatrick, James, Pascanu, Razvan, Rabinowitz, Neil, Veness, Joel, Desjardins, Guillaume, Rusu, Andrei A, Milan, Kieran, Quan, John, Ramalho, Tiago, Grabska-Barwinska, Agnieszka et al., 2017. Overcoming catastrophic forgetting in neural networks. **Proceedings of the national academy of sciences**, 114(13), pp.3521–3526 (cited on page 5).

Lopez-Paz, David and Ranzato, Marc'Aurelio, 2017. Gradient episodic memory for continual learning. **Advances in neural information processing systems**, 30 (cited on page 6).

Luo, Tong, Kramer, Kurt, Goldgof, Dmitry B, Hall, Lawrence O, Samson, Scott, Remsen, Andrew, Hopkins, Thomas and Cohn, David, 2005. Active learning to recognize multiple types of plankton. **Journal of machine learning research**, 6(4) (cited on page 4).

McCloskey, Michael and Cohen, Neal J, 1989. Catastrophic interference in connectionist networks: the sequential learning problem. In: **Psychology of learning and motivation**. Vol. 24. Elsevier, pp.109–165 (cited on pages 2, 5).

Mehta, Sanket Vaibhav, Patil, Darshan, Chandar, Sarath and Strubell, Emma, 2021. An empirical investigation of the role of pre-training in lifelong learning. **Corr** [Online], abs/2112.09153. arXiv: 2112.09153. Available from: https://arxiv.org/abs/2112.09153 (cited on page 5).

Mihaylov, Todor, Clark, Peter, Khot, Tushar and Sabharwal, Ashish, 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In: Ellen Riloff, David Chiang, Julia Hockenmaier and Jun'ichi Tsujii, eds. **Proceedings of the 2018 conference on empirical methods in natural language processing** [Online]. Brussels, Belgium: Association for Computational Linguistics, October, pp.2381–2391. Available from: https://doi.org/10.18653/v1/D18-1260 (cited on pages 4, 5).

Pang, Bo and Lee, Lillian, 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. **Proceedings of the 43rd annual meeting on association for computational linguistics** [Online], Acl '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp.115–124. Available from: https://doi.org/10.3115/1219840.1219855 (cited on page 4).

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei and Liu, Peter J., 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. **Corr** [Online], abs/1910.10683. arXiv: 1910.10683. Available from: http://arxiv.org/abs/1910.10683 (cited on page 6).

Ramírez, Guillem, Lindemann, Matthias, Birch, Alexandra and Titov, Ivan, 2023. Cache & distil: optimising api calls to large language models. **Arxiv preprint arxiv:2310.13561** (cited on pages 1, 2, 3, 4, 5, 6, 15).

Scheffer, Tobias, Decomain, Christian and Wrobel, Stefan, 2001. Active hidden markov models for information extraction. **International symposium on intelligent data analysis**. Springer, pp.309–318 (cited on page 4).

Schröder, Christopher, Niekler, Andreas and Potthast, Martin, 2021. Revisiting uncertainty-based query strategies for active learning with transformers. **Arxiv preprint arxiv:2107.05687** (cited on page 4).

Settles, Burr, 2009. Active learning literature survey (cited on page 3).

Shao, Bo, Doucet, Lorna and Caruso, David R., 2015. Universality versus cultural specificity of three emotion domains: some evidence based on the cascading model of emotional intelligence. **Journal of cross-cultural psychology** [Online], 46(2), pp.229–251. eprint: https://doi.org/10.1177/0022022114557479. Available from: https://doi.org/10.1177/0022022114557479 (cited on page 4).

Stogiannidis, Ilias, Vassos, Stavros, Malakasiotis, Prodromos and Androutsopoulos, Ion, 2023. Cache me if you can: an online cost-aware teacher-student framework to reduce the calls to large language models. **Arxiv preprint arxiv:2310.13395** (cited on pages 1, 2).

Thorne, James, Vlachos, Andreas, Christodoulopoulos, Christos and Mittal, Arpit, 2018. FEVER: a large-scale dataset for fact extraction and VERification. In: Marilyn Walker, Heng Ji and Amanda Stent, eds. **Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)** [Online]. New Orleans, Louisiana: Association for Computational Linguistics, June, pp.809–819. Available from: https://doi.org/10.18653/v1/N18-1074 (cited on pages 4, 5).

Verwimp, Eli, Aljundi, Rahaf, Ben-David, Shai, Bethge, Matthias, Cossu, Andrea, Gepperth, Alexander, Hayes, Tyler L., Hüllermeier, Eyke, Kanan, Christopher, Kudithipudi, Dhireesha, Lampert, Christoph H., Mundt, Martin, Pascanu, Razvan, Popescu, Adrian, Tolias, Andreas S., Weijer, Joost van de, Liu, Bing, Lomonaco, Vincenzo, Tuytelaars, Tinne and Ven, Gido M. van de, 2024. **Continual learning: applications and the road forward**. arXiv: 2311.11908 [cs.LG] (cited on page 2).

A Algorithm

Every new input x_i is initially processed by the student model $S_{i/f}$ to generate a predicted label $\hat{y}_i^S$. Subsequently, there exists an option to acquire the label $\hat{y}_i^T$ from the teacher model T, typically a Large Language Model (LLM), at a cost of $c(x_i)$. The decision to request the teacher model is based on the querying strategy. The returned label $\hat{y}_i$ for each input x_i is either the one provided by the teacher, if requested and the budget permits, or by the student otherwise.

Algorithm 1: Pseudo-code for the neural caching algorithm with datasets (tasks) $Tasks$, the budget for the current task b_{task}, model update frequency f, cost per query c, data from the LLM D_{LLM}, and an initial student S_0.

```
$D_{online} \leftarrow \emptyset$
for $task$ in $Tasks$ do
    for $x_i$ in $X_{online}$ do
        if $i \mod f == 0$ then
            $S_{i/f} \leftarrow$ Update$(D_{LLM}[(i-f):i])$
        end
        $\hat{y}_i \leftarrow S_{i/f}(x_i)$
        if Call_LLM($b, x_i, \hat{y}_i$) and $b_{task} \geq c(x_i)$ then
            $\hat{y}_i \leftarrow$ LLM$(x_i)$
            $b \leftarrow b - c(x_i)$
            $D_{LLM} \leftarrow D_{LLM} \cup \{\langle x_i, \hat{y}_i \rangle\}$
        end
        $D_{online} \leftarrow D_{online} \cup \{\langle x_i, \hat{y}_i \rangle\}$
    end
end
$D_{test} \leftarrow \{\langle x_j, S_{i/f}(x_j) \rangle | x_j \in X_{test}\}$
$Acc_{online} \leftarrow$ Evaluate$(D_{online})$
$Acc_{final} \leftarrow$ Evaluate$(D_{test})$
```

B Architecture diagram

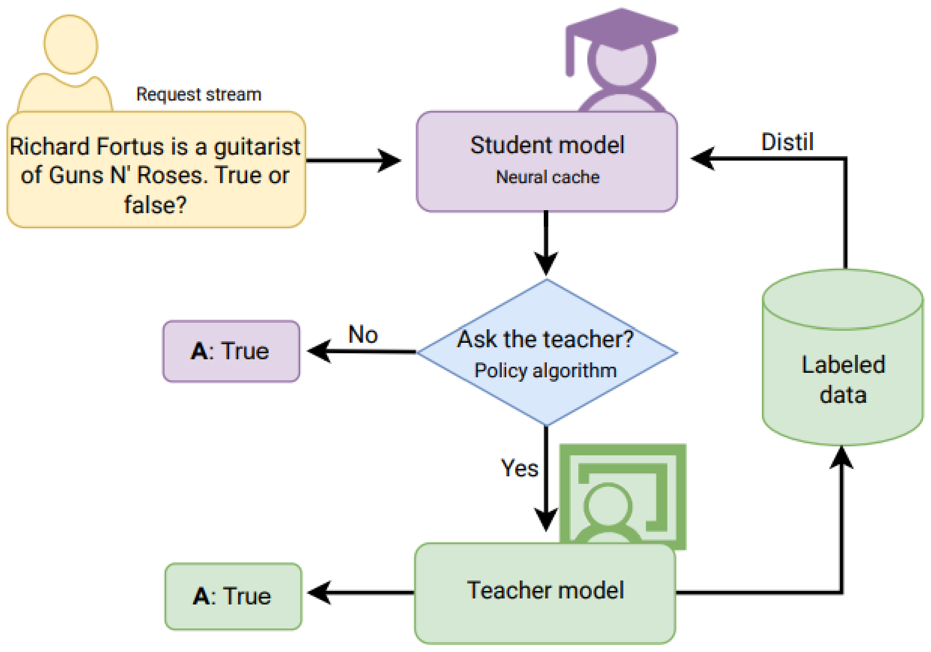

Figure 2 Neural caching framework (Ramírez et al., 2023).