CS 7643 Deep Learning Project Report: Evaluating Context Distillation for LLM Task Adaptation

Joycelyn Ng Kwan Hao Chan Justin Yu Quan Wong Yi Dong Jae
Georgia Institute of Technology
{jng83, kchan303, jwong354, dyi44}@gatech.edu

Abstract

This project evaluates Context Distillation as a task adaptation method for Large Language Models (LLMs), contrasting it with In-Context Learning and Pattern-Based Fine-Tuning. Drawing on the experimental framework of the 2023 'Few-shot Fine-tuning vs. In-Context Learning' study by Mosbach et al.[6], we assess the methods based on in-domain and out-of-domain accuracy, training and evaluation time, training dataset size, and memory requirements.

LLMs adopting In-Context Learning are limited by context length due to memory and computational constraints. Fine-tuning on the context removes the limitation on context length but the model may over-fit on the context. Our research aims to assess whether Context Distillation can surpass In-Context Learning and Pattern-Based Fine-Tuning, by learning from an unconstrained context without over-fitting, for enhanced LLM performance on Natural Language Inference (NLI) tasks.

1. Introduction

1.1. Problem Statement

In-Context Learning enables LLMs to generate responses tailored to specific prompts without additional training or fine-tuning, noted for its ease of use and strong out-of-domain generalization [2]. However, due to memory and computational constraints, its effectiveness may be limited for tasks requiring long contexts, with a typical model handling up to 2048 tokens, or around 32 data samples. Conversely, fine-tuning can extend context length and improve accuracy but risks over-fitting and requires more computational resources [10]. In our study, we adopted Pattern-Based Fine-Tuning from the 'Few-shot Fine-tuning vs. In-Context Learning study to train on a maximum of 128 data samples to limit the training time and computational power. Context Distillation offers a balanced approach, enabling LLMs to handle diverse contexts with less over-fitting while maintaining high accuracy and broad generalization.

1.2. Significance and Sucess Metrics

If we are able to implement Context Distillation with performance superior to In-Context Learning and Pattern-Based Fine-Tuning, Context Distillation could enhance the robustness of LLMs on applications that involve long dialogue conversations, answering questions on lengthy documents or auto-completion of code with knowledge of a large code repository.

For this project, we consider it successful if we manage to get the Context Distillation training working on an LLM, and tweaking the experiment parameters that will allow us to have a fair comparison of the performance between In-Context Learning, Pattern-Based Fine-Tuning and Context Distillation.

1.3. Data Utilized

Our study replicates the setup from the 'Few-shot Fine-tuning vs. In-Context Learning' research, using code and datasets from the https://github.com/uds-lsv/llmft GitHub repository.

Training Dataset

- Multi-Genre Natural Language Inference (MNLI): A corpus designed for training models to perform NLI across multiple genres of text.

Evaluation Datasets

- Multi-Genre Natural Language Inference - Mismatched (MNLI Mismatched): Tests model's generalization across different genres on MNLI mismatched subset.
- Heuristic Analysis for NLI Systems (HANS): Tests model' robustness against heuristic biases.
- Paraphrase Adversaries from Word Scrambling - Quora Question Pairs (PAWS-QQP): Challenges models to identify non-paraphrases despite high lexical similarity.

- Corpus of Linguistic Acceptability - Out Of Distribution (CoLA-OOD): Evaluates model's performance on grammaticality judgment of out-of-distribution examples.

Experimental evaluation in this research is based on GLUE benchmark and dataset as above, which is the current standard for analysing the effectiveness of language model against standard language model performance benchmarks, such as accuracy on identifying specific textual and linguistic relationships.

2. Approach

2.1. In-Context Learning and Pattern-Based Fine-Tuning

Our research implements our In-Context Learning and Pattern-Based Fine-Tuning models as-is from the https://github.com/uds-lsv/llmft GitHub repository with pre-trained model adapted from https://huggingface.co/facebook/opt-125m.

2.2. Context Distillation

Our research references Context Distillation as mentioned in [1]. Hinton et al. [3] provides an equation for Context Distillation loss, given by:

$$\begin{aligned}\text{Loss} = T^2\alpha \cdot \text{KL}\Bigg(\log\text{softmax}\left(\frac{\text{outputs}}{T}\right), \\ \text{softmax}\left(\frac{\text{teacher\_outputs}}{T}\right)\Bigg) \\ + (1-\alpha)\cdot\text{CE}(\text{outputs}, \text{labels})\end{aligned} \quad (1)$$

where KL denotes the Kullback-Leibler (KL) divergence loss, CE denotes the cross-entropy loss, outputs are the logits from the student model, teacher_outputs are the logits from the teacher model, labels are the ground truth labels, T is the temperature scaling parameter, and α is the weighting factor between the soft loss and hard loss.

For our implementation, we define our teacher model to be Pattern-Based Fine-Tuning, while our Context Distillation student model will be trained with the above Context Distillation loss function.

In our baseline experiment, we default T to 1.0 and α to 1.0, which means that the probability distribution output is not softened and the student model will be trained to have a probability distribution output that closely emulates that of the teacher model without direct learning from the ground truth labels of the training examples.

2.3. Challenges and Adaptations

We anticipated difficulties in the initial setup due to the complexity of the models and the novelty of the Context Distillation technique. Indeed, not all went according to plan. The setup process, based on the guidelines and code provided at https://github.com/uds-lsv/llmft, proved to be more time-consuming than expected, in particular wrangling with the deepspeed framework that was used on top of huggingface to train the models. We encountered several issues with the default configurations that necessitated a careful review and correction of the code to align with our project requirements.

We were also limited by the available compute resources, and had to perform all our experiments within the $25 Google Cloud Platform credits that were provided by the instructors. As a result, our experiments were confined to the smallest models that we could use, i.e., OPT-125M, and did not have sufficient credits to perform or re-run some experiments as explained in later sections.

Moreover, grasping the complexity of Context Distillation posed its own set of challenges. As a novel approach with limited precedents, there was a steep learning curve involved. We invested substantial effort in understanding the underlying theoretical framework and in modifying the existing code to accurately model the Context Distillation process. These initial obstacles required persistence and adaptability, laying the groundwork for a more thorough and rigorous application of our methodology.

3. Experiment

3.1. Setup

Commencing with the foundational work, we emulated the experimental framework delineated in the 'Few-shot Fine-tuning vs In Context Learning' paper. Utilizing the repository at https://github.com/uds-lsv/llmft provided us with the requisite code and datasets. To navigate computational constraints, we concentrated our efforts on the smaller 125m model variant, informed by the architecture documented at https://huggingface.co/facebook/opt-125m. Our experiments were facilitated by the computational resources of Google Cloud Platform, ensuring that we had a fail-safe check via Pattern-Based Fine-Tuning, and in context learning. This initial step ensured that our baseline results were consistent with established benchmarks. As an initial step, we first attempted to compare the performance of Pattern-Based Fine-Tuning and In-Context Learning. Additionally, we delved into the relevant literature [1, 4, 8] to understand the mechanisms and theoretical foundations of Context Distillation.

3.2. Result Analysis

We will focus on comparing on Context Distillation (CD) model against the existing In-Context Learning (ICL) and Pattern-Based Fine-Tuning (PBFT) models. Both in-domain (ID) and out-of-domain (OOD) model performance

will be analysed in the following sections. The detailed model performance metrics will also be outlined in . We use the MNLI dataset[9] as the training task for the models below.

3.2.1 Comparison Across Models

Firstly, we evaluate the ID and OOD accuracy of each model. The number of data samples for ICL is limited to 32. For the default PBFT proposed by [6] and CD experiments, default number of sample is 128. Based on our evaluation and past research [6] using 125M parameter OPT model, these parameters demonstrated optimal model performance especially in OOD evaluations. For PBFT and CD, we used batch size of 4 and warm-up ratio at 0.5. We will refer to Table 1 for the following analysis of model accuracy within this section.

PBFT and CD model in-domain accuracy are fairly similar as both models follow similar initial training procedure and configuration. CD introduces the additional distillation procedure from teacher to student model which may lead to a slightly worse performance of CD relative to PBFT. Nevertheless, this relatively small margin of CD and PBFT in-domain accuracy indicates fairly effective distillation from teacher to student model.

However, for OOD accuracy, CD fared significantly worse than ICL and PBFT approach. This is unexpected since in-domain accuracy, we hypothesise CD to demonstrate effective distillation. One possibility is over-fitting during training procedure. A downside of our experiments is the limitation to a small 125M parameter OPT model. Testing CD approach across larger OPT model and allocating more training resource will potentially improve generalisation and OOD performance. It will also provide a better insight on the CD accuracy variations across different OPT model sizes and hyperparameter permutations. ICL accuracy in OOD improved significantly from in-domain which aligns with the findings in 'Few-shot Fine-tuning vs. In-Context Learning' [7]

For other evaluation tasks, the difference between PBFT and CD accuracy in OOD have relatively smaller difference than MNLI evaluation task. For HANS evaluation dataset which evaluates the heuristic reasoning ability of the model [5], CD performed better than PBFT. The additional CD training procedure may have potentially improved CD to better generalise to logical patterns or semantic relationships in the dataset, more effectively than PBFT. ICL has significantly better performance on HANS dataset, as it capture context and linguistic relationships effectively. Similarly, in CoLA dataset which tests model capability to learn grammatical rules, CD does not perform as well as ICL due to its focus on more abstract reasoning skills rather than specific grammatical context.

In PAWS-QQP dataset, CD and PBFT have similar OOD accuracy which are significantly worse than ICL. PAWS-QQP emphasises on the paraphrasing ability of language model which relies on the model understanding of context and word structure prevalent in paraphrasing [11], which tends to be task specific. CD training is targeted to develop more on general reasoning ability, which may explain its worse accuracy in task-specific scenarios in PAWS-QQP task.

3.2.2 Comparison: Training and Evaluation Runtime, Memory Requirements

Training Runtime:

- ICL: There is no training time involved as this method utilizes the pre-trained model as-is, without any adjustments to the weights.
- PBFT: In our experiments, we use a training sample size of 128 across 10 epochs on 1 GPU. The fine-tuning takes 45 minutes on average to complete.
- CD: The training time is longer than pattern-based fine tuning since the student model is the same size as the teacher model. In our experiments with training sample size of 128 across epochs on 1 GPU, it takes 1h 35mins on average.

Our statistics show that CD requires the most training time. Although the reported training times include evaluation time for metrics calculation each epoch, the number of evaluation steps is identical for both PBFT and CD, ensuring the times accurately reflect their respective training durations. There is potential to reduce CD model training time by optimizing the distillation loss calculation or using a smaller training sample size.

Evaluation Runtime:

- ICL: Tends to be slower as the model must process each input with a potentially large context to interpret the examples provided within the prompt.
- PBFT: Generally offers faster evaluation times compared to In-Context Learning because the model applies the learned adjustments directly without the need to process extensive prompt context.
- CD: Evaluation times are expected to be similar to fine-tuning, especially since the model size and complexity are maintained in our case.

Training Memory Requirements:

- ICL: Not relevant since there is no training.

- PBFT: Requires memory for storing the gradients and for the extended datasets used during the training phase. In our experiments for PBFT running on an NVIDIA-T4 GPU, the GPU memory allocated peaked at 47%.

- CD: Since the teacher and student models are the same size in our experimental setup, this method requires significant memory, because we need to hold both models in memory simultaneously during training, along with additional overhead for the distillation process itself (e.g., storing the teacher's outputs to train the student). In our experiments for CD training running on an NVIDIA-T4 GPU, the GPU memory allocated peaked at 76%.

Evaluation Memory Requirements:

- The memory requirements during evaluation are expected to be identical for all three methods, as the same OPT-125M model is used.

3.2.3 Varying Temperature T

For CD, increasing the temperature parameter softens the probability distribution output of a model, reducing the disparity between the higher and lower probabilities. A higher temperature value encourages the student model to learn more nuanced information from the teacher model probability distribution output.

The final step accuracy performance of the CD models with temperature = 1.0 and 2.0 are relatively similar as observed in Figures 1 to 7.

However, as we look across the training steps, the CD model with higher temperature of 2.0 has quicker initial learning for contradiction tasks (2, 4), while the CD model with lower temperature of 1.0 has quicker initial learning for entailment tasks (3, 5).

One possible interpretation for the above would be that contradiction tasks require the model to detect subtler, nuanced distinctions in order to differentiate contradicting from non-contradicting statements. Hence, the CD model with higher temperature of 2.0, having been exposed to a broader range of possibilities and nuances would do well in early learning.

For entailment tasks, the challenge lies in recognizing strong signals of entailment between the premise and the hypothesis. Therefore, a sharper, more confident set of probabilities associated with lower temperature of 1.0 could help the student model quickly latch onto the most relevant features that indicate entailment. This results in better initial performance as the model can effectively focus on the most informative cues without being distracted by less relevant details.

3.2.4 Varying Alpha (α)

For CD, a higher value of alpha places more emphasis on the KL divergence distillation loss rather than the cross entropy loss.

This means the student model is encouraged to align its probability distribution output with that of the teacher model, focusing more on mimicking the teacher's behavior rather than optimizing for the hard labels directly.

An intermediate value of alpha = 0.5 was chosen to strike a balance between learning detailed, soft knowledge from the teacher model and ensuring adequate performance on the direct task as defined by the hard labels.

The final step accuracy performance of the CD models with alpha = 0.5 and 1.0 are relatively similar as observed in Figures 8 to 14.

Interestingly, in the training step range of 200 to 250, there is a peak in accuracy performance for CD model with alpha = 0.5 for contradiction tasks (9, 11), and a dip in accuracy performance for CD model with alpha = 1.0 for entailment tasks (10, 12).

For contradiction tasks, the increase in performance for model with alpha = 0.5 suggests that incorporating some degree of direct classification loss (learning from hard labels) helps the model better identify features that are critical for distinguishing contradiction. Contradiction tasks often require recognizing negations or mutually exclusive statements, which may be more effectively captured when the model also focuses on the ground truth labels rather than solely relying on the teacher's softened outputs.

On the other hand, the decrease in performance for entailment tasks for the same model with alpha = 0.5 could indicate that the direct classification loss introduces noise or guides the student model away from more generalized or subtle patterns captured by the teacher's model. Entailment often requires capturing a broader and more nuanced understanding of compatibility between text statements, which might be better learned through the distillation process, especially when the teacher model has effectively generalized these nuances in its own training.

3.2.5 Varying Training Sample Size

A potential advantage of CD is that it can be more efficient to train than fine-tuning LLMs. In particular, CD may also be used even when the training sample size is small. In this section, we explore the effect of reducing the training sample size on the performance of the LLM when trained using CD.

As a base scenario, we use the same CD model as above, where a training sample size of 128 samples from the MNLI dataset is used. This is compared against training sample sizes of 64, 16 and 2. The mnli_accuracy results are presented in 15.

As the training sample size decreases, the mnli_accuracy metric decreases. The base case with 128 training sample sample size has a final mnli_accuracy of 0.643. However, it is notable that even with a very small training sample size of 2, the model still reasonably well with a final mnli_accuracy of 0.5378. This shows that CD is a reasonable method for training the model even when the amount of data is low, and can be a suitable alternative to In-Context Learning.

It should be noted from 15 that the experimental run for training sample size = 64 was run on training steps than the other runs. This is due to an error in the experimental set up when performing this run, as we did not increase the number of epochs to maintain the same number of training sets after reducing the number of training samples. Due to time limitations, this experiment was not repeated, but the overall trend is still clear.

The MNLI Mismatched evaluation tests are shown in 16 and 17. For the entailment accuracy, there is a clear trend - with more MNLI training samples, the accuracy increases, from 17% accuracy at sample size 2 to 57% at sample size 128. As the model is exposed to more examples, it learns more comprehensive and nuanced patterns about what constitutes entailment. This increase reflects improved generalization as the model sees a wider variety of sentence pairs and contexts, which helps it make better inferences about new, unseen data. However, the contradiction accuracy shows a different trend: the smaller training sample sizes have slightly higher accuracy (around 90% accuracy) compared to the base case of 128 sample size (around 80% accuracy). This is slightly counter intuitive given that the training dataset is also MNLI, and it could hint that at training size of 128, the model is starting to overfit specific features of the training data that do not generalize well to the test set. For example, the model could be picking up on specific patterns or noise within the MNLI dataset that do not actually help in correctly identifying contradictions in a general context.

We further compare the model performance when reducing the training sample size on other out-of-domain evaluation metrics below.

The HANS evaluation tests are shown in 18 and 19. In HANS entailment accuracy, the model shows similar performance when training sample size is 16, 64 or 128, eventually exceeding 90% accuracy. With sample size of 64, the model takes more training steps before the accuracy increases, likely due to the stochastic nature of the learning process. However, with a training sample size of 2, the accuracy does not increase much and seems to peak at 31%, hinting that the model cannot learn sufficiently complex patterns about natural language inference. For HANS overlap accuracy, the model seems to overfit. When the number of training epochs is low, the accuracy increases to near 100%. However, as the training progresses, the accuracy then decreases to almost 0%, likely showing that the model is over-fitting on MNLI dataset which causes it to perform poorly for HANS lexical overlap contradiction test. The over-fitting is not as prominent when the sample size is 2, possibly because the model has not developed the capacity to latch onto misleadingly predictive features of the larger MNLI dataset, which could misguide it when evaluated on HANS.

From 20, the model performance on the CoLA out-of-distribution test is similar for all training sample sizes tested. Interestingly, the model does not need to train on much data to perform well on this dataset. This suggests that the general linguistic knowledge required for the CoLA task is quickly saturated with a small amount of data from MNLI. Since the performance doesn't improve with additional training samples, it also suggests that the MNLI has different task demands from CoLA, and the features learned from further MNLI training does not directly help with the CoLA task.

From 21, the model's PAWS QQP accuracy shows a clear improvement as the training dataset size increases. In particular, when the MNLI training dataset size decreases from 128 to 2, the PAWS QQP accuracy decreases from 0.6691 to 0.2836, clearly showing the benefit of transfer learning. Using a larger portion of the MNLI dataset likely exposes the model to a wider variety of linguistic structures, contexts, and reasoning challenges, helping the model generalize well on the PAWS dataset and allowing it to better handle the complexities of paraphrase identification, particularly when questions have high lexical similarity but differ semantically.

In summary, our exploration reveals that while CD facilitates efficient model training with fewer data, the relationship between training size and model performance is complex and task-dependent. These insights should guide future strategies for model training, particularly in the context of performing transfer learning on LLMs for diverse and challenging NLP tasks.

4. Future Direction

In this section we list down future directions that aim to enhance the efficiency, scope and scalability of our experiments.

4.1. Optimize Context Distillation Training Script

To enhance the computational efficiency of our project, we aim to refine the CD process. Currently, our implementation relies on 'torch.nn.functional' functions like 'log_softmax', which, while effective, may not be the most compute-efficient. Exploring alternative methods to execute these calculations could significantly reduce our computational overhead and accelerate the overall training process. Further, our current methodology requires the teacher

model to be fine-tuned during each CD training session. This is inefficient as it involves repeated computations. We propose to develop a save-and-reload method for the teacher model. By training and fine-tuning the teacher model once and saving its state, we can simply reload this pre-trained model in subsequent CD sessions. This change is expected to streamline the training process, reducing time and computational resources dramatically.

4.2. Leverage Distributed Infrastructure

Our project currently faces limitations due to the availability of only a single GPU on the Google Cloud Platform, which constrains our training capabilities and model scalability. To address this, we are considering the adoption of a distributed training infrastructure that would allow us to provision multiple GPUs. This enhancement would not only speed up the training process but also enable us to experiment with larger models.

Currently, our experiments are conducted on smaller models like the OPT-125M. Access to a more robust computing infrastructure would allow us to test our hypotheses and models on larger architectures, such as OPT-1.3B or larger, providing insights into whether the observed trends are consistent across different scales and complexities of models.

4.3. Tune Additional Hyperparameters

In our current setup, we have primarily focused on adjusting a limited set of hyperparameters, specifically alpha and temperature in the CD calculations, along with the training sample size. To further optimize our model's performance and adaptability, a broader range of hyperparameters can be explored, for example varying the batch size, learning rate, and potentially other scheduler and optimiser settings.

Expanding the scope of hyperparameter tuning will provide a more comprehensive understanding of how different settings impact model performance across various tasks. This approach not only enhances the model's effectiveness but also contributes to the robustness and generalizability of the CD process.

5. Conclusion

This study has embarked on an in-depth examination of three distinct task adaptation methods for Large Language Models (LLMs): In-Context Learning, Pattern-Based Fine-Tuning, and Context Distillation. Each method was compared using in-domain and out-of-domain accuracy, training and evaluation times, and memory requirements.

Our findings indicate that Context Distillation offers a promising middle ground between In-Context Learning and pattern-based fine tuning.

Context Distillation with a smaller training sample size was also shown to be reasonably effective, performing particularly well for specific tasks. This offers a compelling alternative for few-shot In-Context Learning, with the added advantage of not being limited by the small context window of In-Context Learning.

Future research should explore the scalability of Context Distillation across more diverse datasets and in more complex application scenarios. Additionally, further refinement of the distillation techniques could enhance their efficiency and applicability, making them more accessible for a wider range of LLM applications.

References

[1] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861, 2021. 2

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. 1

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 2

[4] Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen R. McKeown. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models. *CoRR*, abs/2212.10670, 2022. 2

[5] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. 3

[6] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *CoRR*, abs/2305.16938, 2023. 1, 3

[7] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[8] Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *CoRR*, abs/2209.15189, 2022. 2

[9] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. 3

[10] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. *CoRR*, abs/2006.05987, 2020. 1

[11] Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3

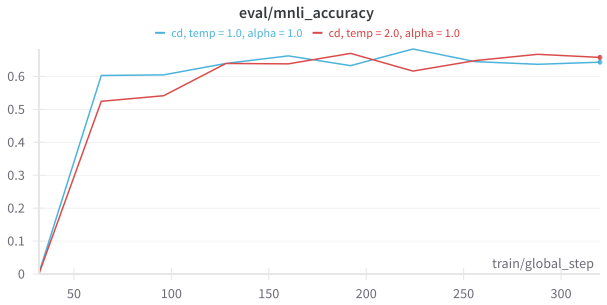

Figure 1. MNLI Accuracy (Temp = 1.0, 2.0)

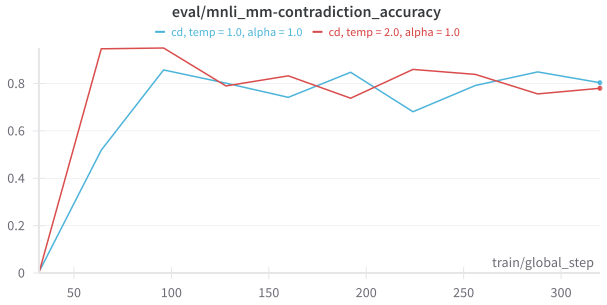

Figure 2. MNLI MM Contradiction Accuracy (Temp = 1.0, 2.0)

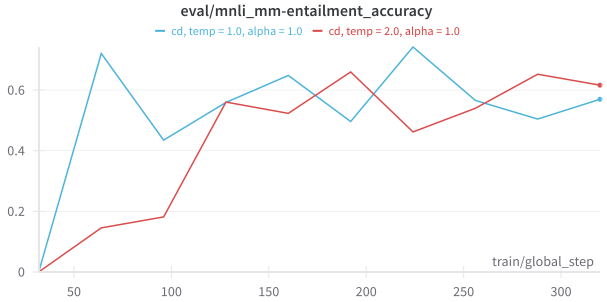

Figure 3. MNLI MM Entailment Accuracy (Temp = 1.0, 2.0)

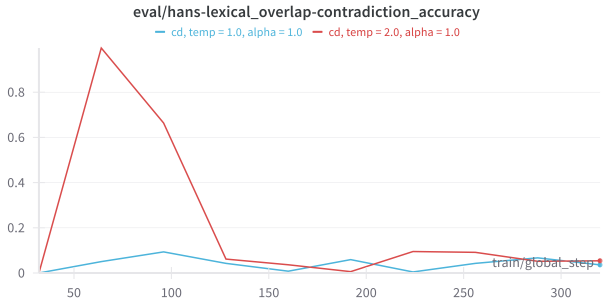

Figure 4. HANS Lexical Overlap Contraction Accuracy (Temp = 1.0, 2.0)

Model (No. of Data Samples)	ID - MNLI	OOD - MNLI MM	OOD - Hans	OOD - Paws QQP	OOD - Cola
ICL (32)	0.5212	0.9974	0.9999	0.9938	0.9753
PBFT (128)	0.6986	0.9972	0.7176	0.7164	0.688
CD (128)	0.6452	0.6385	0.7844	0.709	0.686

Table 1. In-Domain (ID) and Out-of-Domain (OOD) Accuracy Across Models

No. of Data Samples	ID - MNLI	OOD - MNLI MM	OOD - Hans	OOD - Paws QQP	OOD - Cola
2	0.5418	0.8366	0.938	0.4835	0.006
16	0.5200	0.9994	0.9992	0.9998	0.8765
32	0.5212	0.9974	0.9999	0.9938	0.9753

Table 2. In-Context Learning In-Domain (ID) and Out-of-Domain (OOD) Accuracy across Context Sample Size

No. of Data Samples	ID - MNLI	OOD - MNLI MM	OOD - Hans	OOD - Paws QQP	OOD - Cola
16	0.5031	0.1285	0.164	0.5037	0.688
32	0.5544	0.6899	0.9862	0.7164	0.6725
64	0.6322	0.5539	0.9957	0.7179	0.6899
128	0.6986	0.7176	0.9972	0.7164	0.688

Table 3. Pattern-Based Fine-Turning In-Domain (ID) and Out-of-Domain (OOD) Accuracy across Training Sample Size

Temperature	Alpha	ID - MNLI	OOD - MNLI MM	OOD - Hans	OOD - Paws QQP	OOD - Cola
1	0.5	0.6433	0.5697	0.9608	0.6691	0.688
1	1	0.6452	0.6385	0.7844	0.709	0.686
2	0.5	0.6822	0.8227	0.9466	0.7149	0.688
2	1	0.6578	0.6162	0.9434	0.6957	0.688

Table 4. Context Distillation In-Domain (ID) and Out-of-Domain (OOD) Accuracy across Temperature and Alpha

No. of Data Samples	ID - MNLI	OOD - MNLI MM	OOD - Hans	OOD - Paws QQP	OOD - Cola
2	0.5409	0.1473	0.2386	0.2836	0.686
16	0.5566	0.2322	0.9292	0.4815	0.686
32	0.4807	0.004	0.0054	0.288	0.686
64	0.6463	0.6214	0.9998	0.7105	0.688
128	0.6452	0.6385	0.7844	0.709	0.686

Table 5. Context Distillation In-Domain (ID) and Out-of-Domain (OOD) Accuracy across Training Sample Size

Student Name	Contributed Aspects	Details
Joycelyn Ng	Implementation and Analysis	Guided the design of the code to perform the context distillation training. Improved code to better report the results of the teacher and student models. Ran the context distillation experiments, analysed the results and wrote the relevant report sections.
Kwan Hao Chan	Implementation and Analysis	Performed experimental runs on In-Context Learning and pattern-based fine tuning. Collated the results, provided analysis, and wrote relevant report sections.
Justin Yu Quan Wong	Implementation and Analysis	Wrote code to perform the context distillation training. Guided the experimental setup, ran the context distillation experiments, analysed the results and wrote the relevant report sections.
Yi Dong Jae	Implementation and Analysis	Performed experimental runs on In-Context Learning. Collated the results, provided analysis, and wrote relevant report sections.

Table 6. Contributions of Team Members.

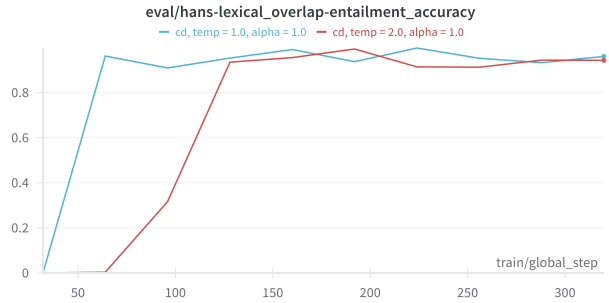

Figure 5. HANS Lexical Overlap Entailment Accuracy (Temp = 1.0, 2.0)

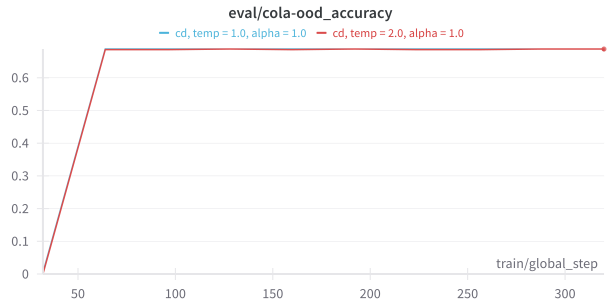

Figure 6. CoLA OOD Accuracy (Temp = 1.0, 2.0)

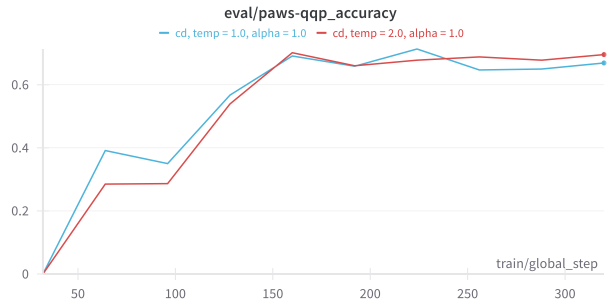

Figure 7. PAWS-QQP Accuracy (Temp = 1.0, 2.0)

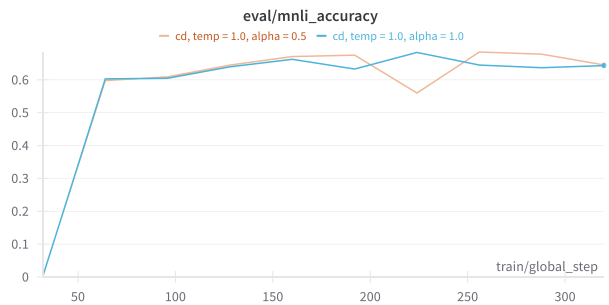

Figure 8. MNLI Accuracy (Alpha = 0.5, 1.0)

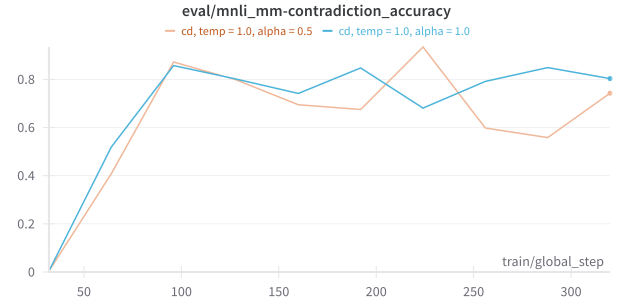

Figure 9. MNLI MM Contradiction Accuracy (Alpha = 0.5, 1.0)

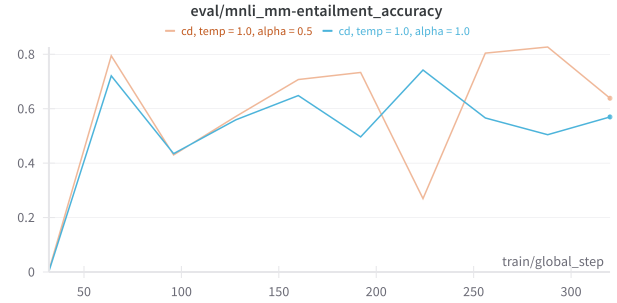

Figure 10. MNLI MM Entailment Accuracy (Alpha = 0.5, 1.0)

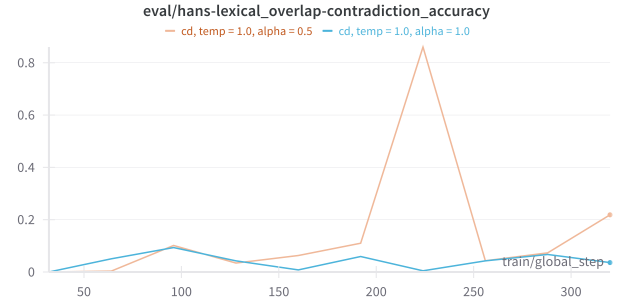

Figure 11. HANS Lexical Overlap Contraction Accuracy (Alpha = 0.5, 1.0)

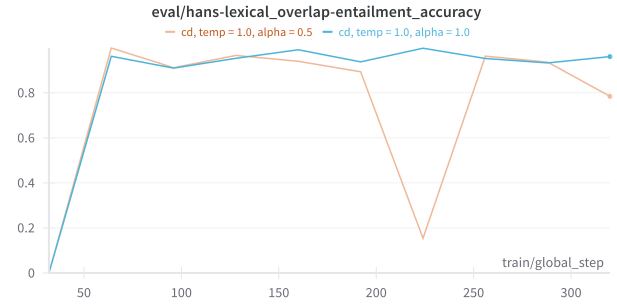

Figure 12. HANS Lexical Overlap Entailment Accuracy (Alpha = 0.5, 1.0)

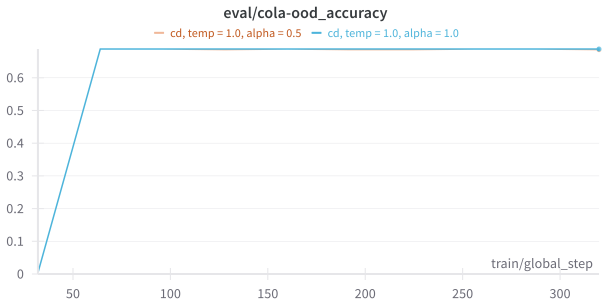

Figure 13. CoLA OOD Accuracy (Alpha = 0.5, 1.0)

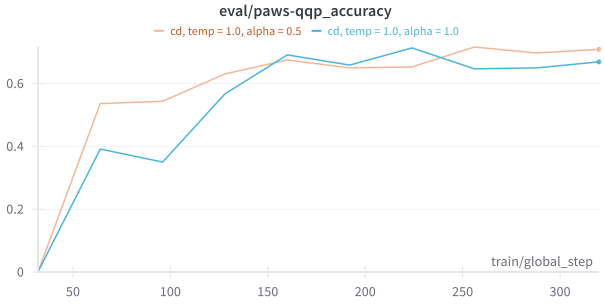

Figure 14. PAWS-QQP Accuracy (Alpha = 0.5, 1.0)

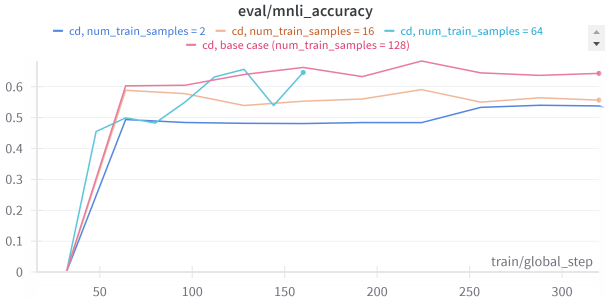

Figure 15. MNLI Accuracy (Training Sample Size = 2, 16, 64, 128)

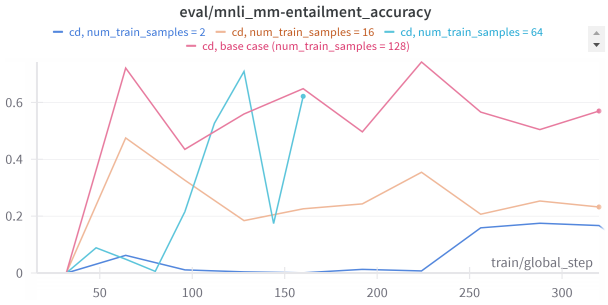

Figure 16. MNLI MM Entailment Accuracy (Training Sample Size = 2, 16, 64, 128)

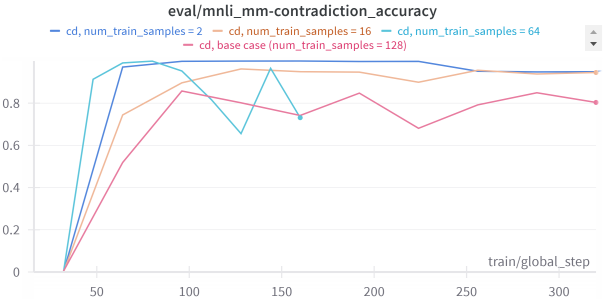

Figure 17. MNLI MM Contradiction Accuracy (Training Sample Size = 2, 16, 64, 128)

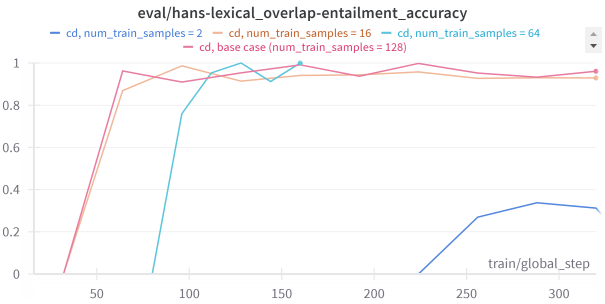

Figure 18. HANS Lexical Entailment Accuracy (Training Sample Size = 2, 16, 64, 128)

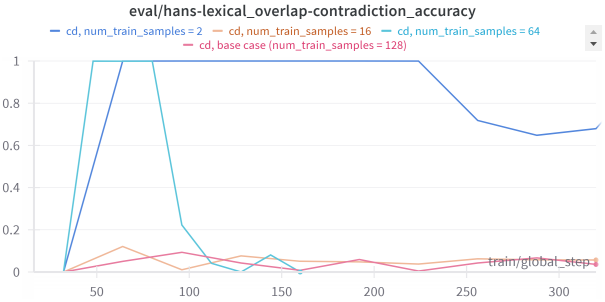

Figure 19. HANS Lexical Contradiction Accuracy (Training Sample Size = 2, 16, 64, 128)

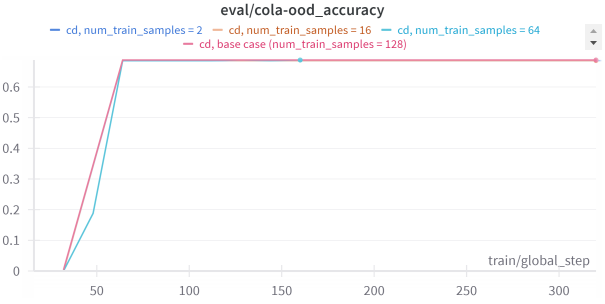

Figure 20. CoLA OOD Accuracy (Training Sample Size = 2, 16, 64, 128)

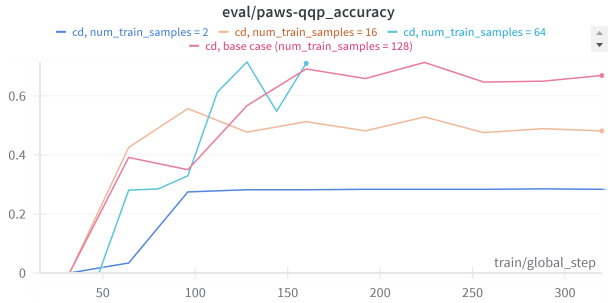

Figure 21. PAWS QQP Accuracy (Training Sample Size = 2, 16, 64, 128)