CLAfICLe: Cross Lingual Adaptation for In-Context Learning

Giulio Starace
University of Amsterdam / Amsterdam, The Netherlands
giulio.starace@gmail.com

Abstract

This document is a supplement to the general instructions for *ACL authors. It contains instructions for using the LaTeX style files for ACL conferences. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used both for papers submitted for review and for final versions of accepted papers.

1 Introduction

in-context learning (ICL) Decoder-only transformer (DOT). contributions

- successfully apply WECHSEL to GPT2 Large, release the checkpoints which did not exist
- More complete evaluation of WECHSEL in the GPT2 setting
- propose method for preserving FT when performing cross-lingual adaptation
- formalize concept of vessel adapters with targeted distillation, a form of post-hoc disentanglement. We call it PHoDiVA (**P**ost **Ho**c **Di**sentanglement via **V**essel **A**dapters).

2 Related Work

todo

3 Methods and Models

3.1 MetaICL

Due to the complete lack of prompting/instruction templates in non-English languages, we rely on MetaICL (Min et al., 2022), which circumvents the need for prompt/instruction templates at train-time and test-time. With MetaICL, a pretrained DOT is fine-tuned by concatenating k examples of input-output pairs from a variety of tasks and feeding this as input to the model. The final input-output pair is truncated such that only the input is shown, and the model is trained to predict the output using a negative log-likelihood objective. The trained model is then able to generalise to unseen tasks presented in the same way by utilizing the k shots provided in the context. We refer to this model simply as *MetaICL*.

3.2 WECHSEL

To adapt a monolingual DOT from a source language to a target language we employ WECHSEL (Minixhofer et al., 2022), which has shown success in adapting the small variant of GPT2 (117M parameters) to a number of target languages. WECHSEL works by retraining the tokenizer into the target language and re-initializing the transformer embedding layers such that the target embeddings are semantically similar to the source embeddings. This is done by leveraging existing parallel multilingual static word embeddings. As done by de Vries et al. (2021), after re-initialization, additional causal language modeling (CLM) is performed in the target language to account for syntactical differences. Applying WECHSEL to MetaICL, we obtain what we refer to as *MetaICL-geWECHSELt*.

3.3 Adapters

Because we are interested in adapting a fine-tuned DOT (MetaICL), we hypothesize that the additional CLM at the end of WECHSEL can lead to catastrophic forgetting of the fine-tuning. Furthermore, we hypothesize that the fine-tuning may contain language-specific information, entangled with the task information relevant to the fine-tuning objective. To address this issue, inspired by MAD-X (Pfeiffer et al., 2020b) we train a “task adapter” on the same ICL objective and data as MetaICL with a GPT2 base, obtaining an “ICL-adapter”, which we refer to as *MetaICLA*. Adapters introduce “bottleneck” dense layers at each transformer layer of their base. The adapter is trained on a particular objective while the base is kept frozen, allowing for parameter-efficient and modular fine-tuning. These dense layers consist in a down matrix $\mathbf{W}_{down}$, projecting the hidden states into a lower dimension $d_{bottleneck}$, a non-linearity f, which is applied to this projection and an up matrix $\mathbf{W}_{up}$ that projects back to the original dimension:

$$\mathbf{h} \leftarrow \mathbf{W}_{up} f(\mathbf{W}_{down}\mathbf{h}) + \mathbf{r}, \quad (1)$$

where r is a residual connection. Various configurations of the above exist. Having separated the task-specific information, we apply WECHSEL to the GPT2 base, obtaining what we refer to as *GPT2-geWECHSELt*. Adding MetaICLA to GPT2-geWECHSELt, we obtain a model theoretically capable of ICL on the target language, *GPT2-geWECHSELt+MetaICLA*.

3.4 PHoDiVA

To address situations where repeating fine-tuning is not permissible, either because the data is not released, the process too complicated or the compute simply not available, we propose PHoDiVA

todo

3.5 Sandwich

todo

4 Experimental Setup

For our work, we use the `pfeiffer` configuration from AdapterHub (Pfeiffer et al., 2020a).

Min et al. (2022) train a number of variants, releasing checkpoints however only for the large variant (774M parameters) of GPT2. We base the rest of our models on the same GPT2 variant and use the "high resource to low resource" direct MetaICL checkpoint as we consider this to be the most realistic.

4.1 Training

4.1.1 CrossFit and UnifiedQA

4.1.2 OSCAR

4.2 Evaluation

4.2.1 Multi-lingual Multi-task Benchmark

5 Results and Discussion

Fig. 2 shows the performance of GPT2 after around 1k steps of training, evaluated intrinsically in terms of perplexity. For both French and German, we see perplexity decrease to sub-50 values, with the French model reaching a perplexity of ≈ 28. Both models are clearly underfit, still monotonically decreasing by the end of the training. These observations are roughly in-line with Minixhofer et al. (2022)'s findings for smaller variants of GPT2, although we train for much less time and hence are left with higher perplexities. While we believe our preliminary results suggest WECHSEL scales well to larger models in terms of intrinsic evaluation, future work may wish to investigate whether this holds for longer training times. The rest of our work considers, among other questions, the robustness of WECHSEL via extrinsic evaluation on downstream tasks performed by MetaICL.

Fig. 3 shows the performance on each dataset of our benchmark for the two baseline models, MetaICL and Sandwich. As summarized in Table 1, Sandwich performs roughly on-par with MetaICL on both target languages, respectively with scores of 0.317 and 0.322 in French and German compared to MetaICL's score of 0.327 in English. We note generally low scores across all tasks. This is particularly perplexing in the case of MetaICL, scoring around 0.1 points less than with the evaluation ensemble used by Min et al. (2022), where the same checkpoint was reported scoring 0.417 in the worst case (a 25 % decrease). While similar values are reached in certain tasks in our benchmark (e.g. most of XGLUE and WINO-X), it is unclear what the origin of this discrepancy is, whether due to differences in evaluation implementation or difficulty of the tasks. Given that Min et al. (2022) simply report macro-averaged scores, it is impossible to verify the latter. Nevertheless, our results suggest that Sandwich-like solutions may be satisfactory for transferring performance from English to other languages given the surprisingly closeness of the scores. The decision between using Sandwich or "properly" adapted models with the same capabilities then becomes an economic one in terms of the cost of API calls (for the former) versus the cost of inference plus training (for the latter).

Fig. 4 shows the difference in performance on each dataset of our benchmark between the proposed models and Sandwich. In general, we observe that the proposed models underperform across almost all tasks in both French and German, with the trends aligning at a task-level (e.g. all models underperform on QAM, by roughly the same amount). As reported in Table 1, the best of our proposed models is MetaICL-geWECHSELt,

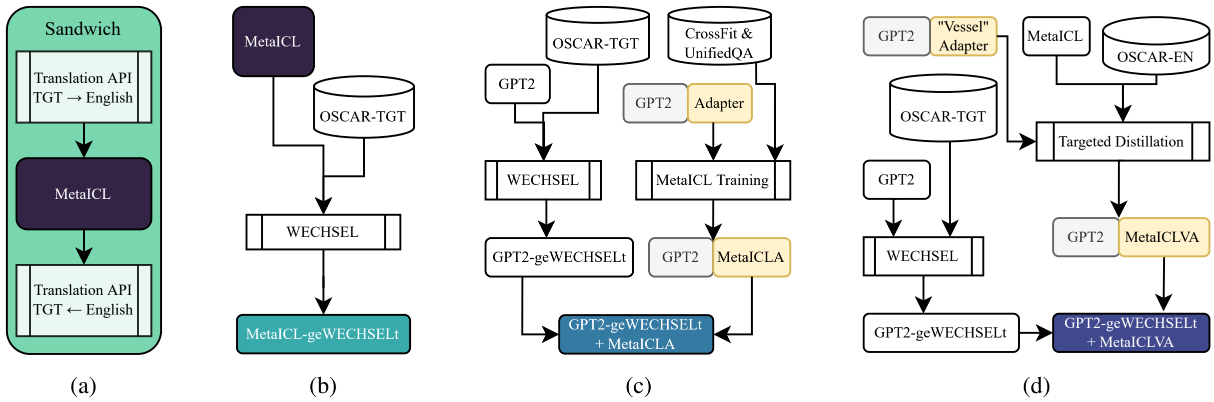

Figure 1: Overview of each of the models evaluated in one of the two TGT languages (French or German). The baseline Sandwich model (a) sandwiches MetaICL (Min et al., 2022) (which we separately evaluate only in English) between two complementary translation API calls. MetaICL-geWECHSELt (b) is the result of applying WECHSEL (Minixhofer et al., 2022) to MetaICL. GPT2-geWECHSELt+MetaICLA combines MetaICLA, an adapter trained on the MetaICL dataset and objective, with a TGT-language GPT2 base obtained via WECHSEL. GPT2-geWECHSELt+MetaICLVA does the same, except MetaICLVA is trained via targeted distillation with supervision provided by MetaICL. For more details, refer to section 3.

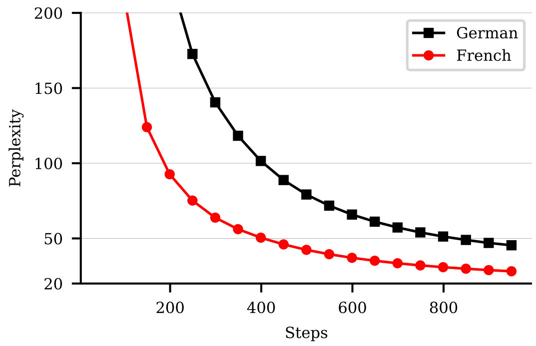

Figure 2: Perplexity on the held out set when performing the recommended CLM training after WECHSEL language-adaptation of GPT2. A step corresponds to an optimizer update. We evaluate every 50 steps.

which underperformed Sandwich by roughly 0.02-0.03 points. This undermines the motivation for the other two models, which were designed to avoid catastrophic forgetting by separating language and ICL capabilities via adapters. The results suggest that the tradeoff between catastrophic forgetting and needing to train ICL-adapters leans in favour of the former in this compute regime. In this sense, we can conclude that WECHSEL does not suffer tremendously due to catastrophic forgetting when adapting fine-tuned DOTs such as the MetaICL variant of GPT2.

Our work is mainly limited by its preliminary nature. Apart for considering more appropriate (*sc.* larger) compute scales, future work could investigate training ICL-adapters more thoroughly, for example by performing hyperparameter optimization or incorporating more recent adapter research such as AdapterDrop (Rücklé et al., 2021), AdapterFusion (Pfeiffer et al., 2021) or Hyper-X (Üstün et al., 2022).

We are also interested in a more complete treatment of PhoDiVA. For instance, future work could explore different forms of student-teacher distillation, consider other forms of loss criterions and/or take inspiration from similar solutions such as Khrulkov et al. (2021)'s work on generative models. We believe work in this direction could benefit from simplifying the problem setting first, by considering a smaller, encoder-only transformer fine-tuned on a single downstream task on a single-language.

Other future work may consider different adaptation approaches that have recently emerged. For example, Marchisio et al. (2022)'s Mini-Model adaptation has yet to be tested on decoder-only transformers, and it would be interesting to see how it compares to WECHSEL in this regard. Other, slightly more distant approaches such as meta-learning a-la X-MAML (Nooralahzadeh et al., 2020) may provide different results.

Perhaps a clear limitation of this direction of research is that the setting remains monolingual. Future work could explore whether it is possible to adapt a monolingual model to multiple languages simultaneously, and how such adaptations would compare to monolingual-to-monolingual adaptation in terms of resources and performance. Finally, undermining all of this work is our restriction to

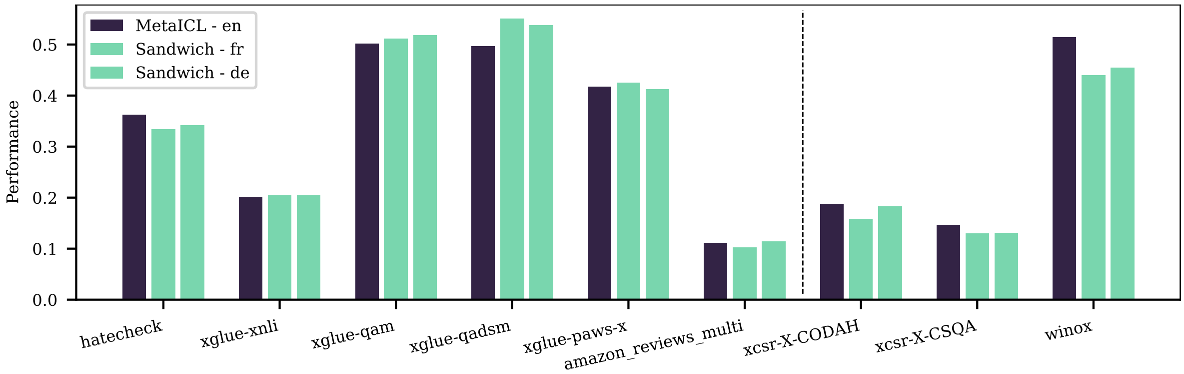

Figure 3: Performance (max is 1) on a particular language dimension of our multi-task benchmark of our two baseline models, MetaICL and Sandwich. The dashed line separates whether a given task uses accuracy (left) or F1-score (right) as the performance metric.

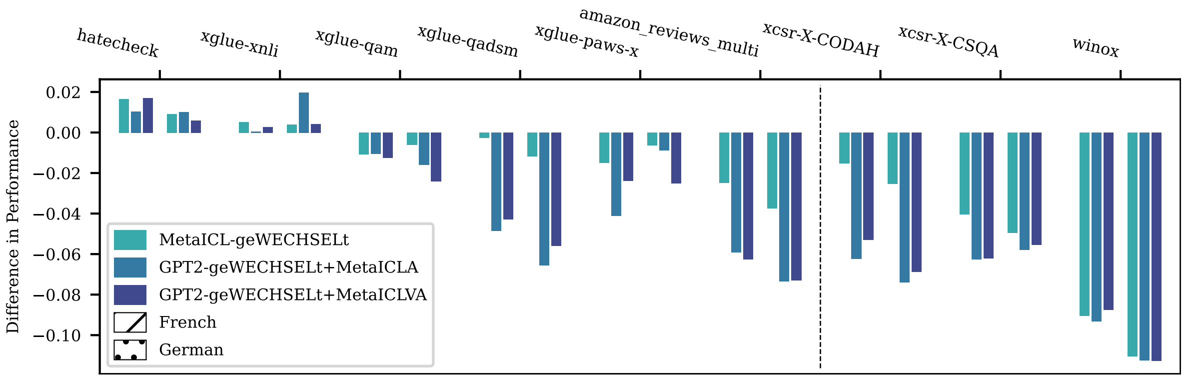

Figure 4: Performance gap on our multi-task benchmark between each of the language-adapted models and the "Sandwich" baseline. Positive values indicate that the adapted models are outperforming the baseline, while negative values indicate the reverse. The dashed line separates whether a given task uses accuracy (left) or F1-score (right) as the performance metric.

results on a single random seed. Future work with more seeds and more compute would be necessary to draw more definitive conclusions.

Table 1: Average performance (max is 1) across the datasets from our multi-task benchmark for the models considered in this work. We use "W" as a shorthand for "geWECHSELt". We report average difference in performance for each proposed alternative to Sandwich. Negative values indicate underperformance compared to Sandwich.

	en	fr	de
MetaICL	0.327	-	-
Sandwich	-	0.317	0.322
Difference in Performance w.r.t. Sandwich			
MetaICL-W	-	-0.020	-0.026
GPT2-W+MetaICLA	-	-0.041	-0.042
GPT2-W+MetaICLVA	-	-0.036	-0.045

6 Conclusion

We explore the problem of language-adapting a monolingual DOT previously fine-tuned to perform in-context learning. To this end, we stress test the current SoTA adaptation method, WECHSEL, scaling to previously untested model sizes, applying it a fine-tuned variant of GPT2 (MetaICL) and evaluating extrinsically on a multi-task benchmark. While we find that WECHSEL successfully scales to larger model sizes, we find that at our compute regime, WECHSEL-adapted MetaICL underperforms compared to simply sandwiching the English model between translation API calls. We experiment with separating ICL fine-tuning and language adaptation to address potential catastrophic forgetting through the use of Adapters, but find these approaches unsuccessful. In doing so, we propose PHoDiVA, a novel method for post-hoc disentanglement through vessel adapters. We share PHoDiVA in this rudimentary form as a starting point for future work in this direction.

References

Wietse de Vries, Martijn Bartelds, Malvina Nissim, and Martijn Wieling. 2021. Adapting Monolingual Models: Data can be Scarce when Language Similarity is High. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4901–4907, Online. Association for Computational Linguistics.

Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. 2021. Disentangled Representations from Non-Disentangled Models.

Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2022. Mini-Model Adaptation: Efficiently Extending Pretrained Models to New Languages via Aligned Shallow Training.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to Learn In Context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-Shot Cross-Lingual Transfer with Meta Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the Efficiency of Adapters in Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, Gertjan van Noord, and Sebastian Ruder. 2022. Hyper-X: A Unified Hypernetwork for Multi-Task Multilingual Transfer.

7 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

A Example Appendix

This is an appendix.