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 ("shots") 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 from a number of possible options. The trained model is then generalises to unseen tasks presented in the same way by utilizing the k shots provided in the context. We refer to this model as *MetaICL*.

3.2 Sandwich

As a baseline, we consider the obvious solution of simply translating input in the target language to English, feeding the translation to MetaICL, and translating the output back to the target language. We refer to this model as *Sandwich*. We make use of Google's Cloud Translation AI API[1].

3.3 WECHSEL

Aside from translation API calls, 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*.

3.4 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

[1] https://cloud.google.com/translate

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:

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

where r is a residual connection. 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 in the target language, *GPT2-geWECHSELt+MetaICLA*.

3.5 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. Here, instead of repeating ICL fine-tuning, we leverage the fine-tuned MetaICL checkpoint, using it as a teacher in a modified student-teacher offline distillation (Hinton et al., 2015) setup. More specifically, before WECHSEL adaptation, we add a "vessel" adapter to a (frozen) GPT2 base, and then perform CLM in the source language (English). Vessel adapters are exactly the same as task adapters, except that they act as a "vessel" for distilled capabilities rather than as additional parameters for fine-tuning. Rather than predicting the actual next word, the adapter is trained to predict the next word greedily sampled from the teacher. The idea is to overfit the adapter to the teacher outputs (hence the greedy sampling). Because the GPT2 base is frozen and theoretically shares the original language modeling capabilities of the teacher, we hypothesize that this "targeted distillation" can disentangle the fine-tuned capabilities into the vessel adapter. We use the CLM objective because of the constraint to keep the distillation process as simple as possible, so to make it advantageous over repeating a potentially complex fine-tuning process. The only constraint of this method is that the adapter base is the same pretrained base that was fine-tuned into the teacher. When using MetaICL as the teacher, we refer to the resulting vessel adapter as *MetaICLVA*. Like in section 3.4, after applying WECHSEL to a GPT2 base, we can then combine the language-adapted base and MetaICLVA to obtain *GPT2-geWECHSELt+MetaICLVA*, another model theoretically capable of ICL in the target language.

4 Experimental Setup

We use the PyTorch Lightning Python framework (Falcon and The PyTorch Lightning team, 2019) to implement our work. Because we envision the direct application of this work to be most useful to smaller companies and start-ups, we limit our compute to a single 40GB NVIDIA A100 GPU and run jobs for a maximum of 24 hours. Our code is available on GitHub[2].

4.1 Models

There exist various possible adapter setups, specifying different configurations of the up and down weight matrices, non-linearity and residual connection, among other settings. For our work, we use the `pfeiffer` configuration from AdapterHub (Pfeiffer et al., 2020a).

Regarding MetaICL, Min et al. (2022) train a number of variants, releasing checkpoints however only for variants fine tuning the large version of GPT2 (774M parameters). We base the rest of our models on the same GPT2 version and use the "high resource to low resource" direct MetaICL checkpoint as we consider this to be the most realistic. We make use of the HuggingFace Transformers (Wolf et al., 2020) implementation of GPT2 throughout.

4.2 Evaluation

For assessing ICL, we reimplement the same evaluation setup as Min et al. (2022), evaluating a given model on a suite of tasks, prepending the input with $k = 16$ training examples sampled randomly for each task. For our evaluation metrics, like Min

[2] https://github.com/thesofakillers/CLAfICLe

Table 1: The datasets constituting our ICL benchmark. Most originate from pre-existing benchmarks, namely XGLUE (Liang et al., 2020) and (Lin et al., 2021).

Dataset	(Origin)	Collection
HateCheck	(Röttger et al., 2021)	-
XNLI	(Conneau et al., 2018)	XGLUE
QAM	(Liang et al., 2020)	XGLUE
QADSM	(Liang et al., 2020)	XGLUE
PAWS-X	(Yang et al., 2019)	XGLUE
MARC	(Keung et al., 2020)	-
X-CODAH	(Lin et al., 2021)	XCSR
X-CSQA	(Lin et al., 2021)	XCSR
Wino-X	(Emelin and Sennrich, 2021)	-

et al. (2022), we use F1 for tasks where the label options change across examples, and accuracy for tasks where the label options are always the same. Because the evaluation benchmark used by Min et al. (2022) is limited to English, we develop our own multilingual multi-task benchmark spanning 9 different tasks across 3 languages (English, German, and French). Our benchmark design is restricted to tasks that can be handled by the MetaICL framework, namely multi-class, single-label tasks. To enable complete comparisons across languages, we also restrict our benchmark to only contain language-parallel datasets. Correspondingly, we list the ICL benchmark datasets in Table 1.

To evaluate the successful application of WECHSEL to GPT2, we use the same process as Minixhofer et al. (2022), namely measuring perplexity on a held out test set. For all language modeling, we use the original release of the OSCAR corpus (Ortiz Suárez et al., 2020).

4.3 Training

When performing CLM training, due to our limited compute, we heed the advice of Geiping and Goldstein (2022) and pack as samples into 1024-token sequences (the maximum length possible) by separating them with EOS tokens, so to minimize the number of padding tokens and maximize GPU utilisation. With this we are able to fit a batch size of 2 into memory, while actually presenting the model with more than two examples per batch in most cases[3]. We achieve a virtual batch size of 512 by accumulating gradients over 256 steps. We employ single-epoch training (Komatsuzaki, 2019) on a total of 600M tokens, which we estimate to be the number of tokens consumed by our model in a single epoch by running a profiling run on a smaller download. Based on the information in Geiping and Goldstein (2022) and Minixhofer et al. (2022), we decide to use Adam (Kingma and Ba, 2015) with a linear warmup for the first half of training to a peak learning rate of 5e-4, followed by cosine annealing to 0 by the end of training. When performing targeted distillation for MetaICLVA, we reduce the linear warmup to the first 10% of training to help with our voluntary overfitting. As suggested by Izsak et al. (2021), to maximize training time, we evaluate on only 0.5% of the data, logging every 50 steps.

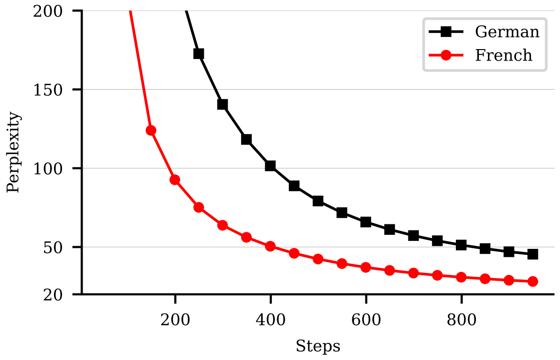

Figure 1: Perplexity on the held out set when performing the recommended CLM training after WECHSEL language-adaptation of GPT2. A step corresponds to an optimizer update. We evaluate every 50 steps.

For the ICL training necessary for MetaICLA, we modify Min et al. (2022)'s implementation so to work with adapters. In particular, we use their HR→LR training mixture, which consists of 61 tasks sourced from the CrossFit (Ye et al., 2021) and UnifiedQA (Khashabi et al., 2020) benchmarks.

5 Results and Discussion

Fig. 1 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 evalua-

[3] This technique is also suggested by HuggingFace in their CLM tutorial: https://huggingface.co/course/chapter7/6.

tion, 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. 2 shows the performance on each dataset of our benchmark for the two baseline models, MetaICL and Sandwich. As summarized in Table 2, 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. 3 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 2, the best of our proposed models is MetaICL-geWECHSELt, 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, use beam search rather than greedy sampling 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 results on a single random seed. Future work with more seeds and more compute would be necessary to draw more definitive conclusions.

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,

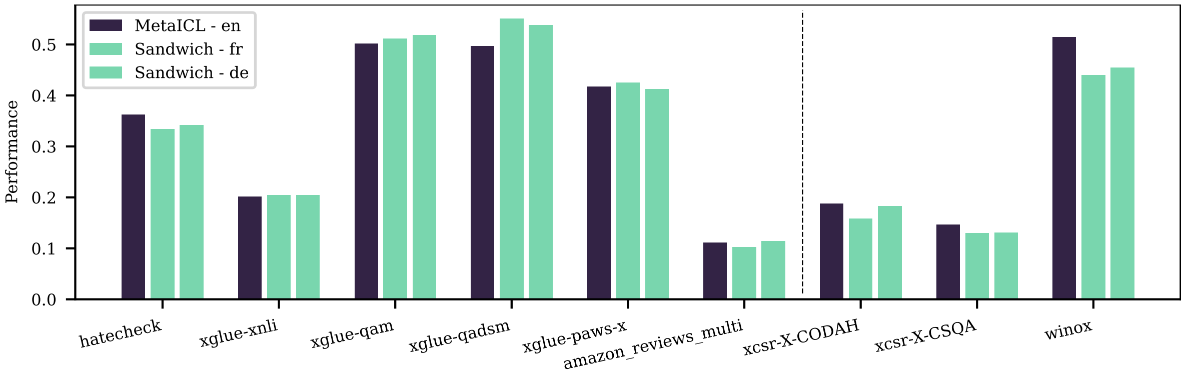

Figure 2: Performance (max is 1) on a particular language dimension of our multi-task benchmark of our two baseline models, MetaICL and Sandwich. The dashed line separates whether a given task uses accuracy (left) or F1-score (right) as the performance metric.

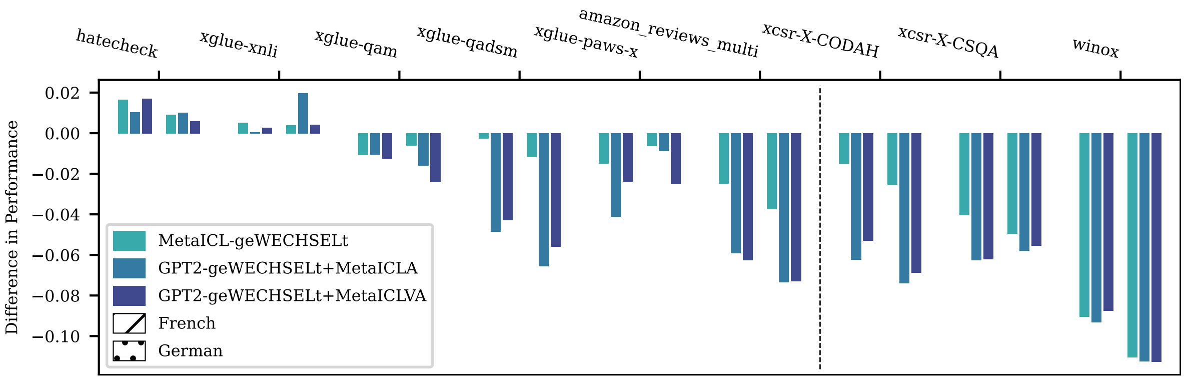

Figure 3: Performance gap on our multi-task benchmark between each of the language-adapted models and the "Sandwich" baseline. Positive values indicate that the adapted models are outperforming the baseline, while negative values indicate the reverse. The dashed line separates whether a given task uses accuracy (left) or F1-score (right) as the performance metric.

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

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. *arXiv:1809.05053 [cs]*.

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.

Denis Emelin and Rico Sennrich. 2021. Wino-X: Multilingual Winograd Schemas for Commonsense Reasoning and Coreference Resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Jonas Geiping and Tom Goldstein. 2022. Cramming: Training a Language Model on a Single GPU in One Day.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015.

Table 2: 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

Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to Train BERT with an Academic Budget.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The Multilingual Amazon Reviews Corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing Format Boundaries With a Single QA System.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Aran Komatsuzaki. 2019. One Epoch Is All You Need.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A New Benchmark Datasetfor Cross-lingual Pre-training, Understanding and Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common Sense Beyond English: Evaluating and Improving Multilingual Language Models for Commonsense Reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1274–1287, Online. Association for Computational Linguistics.

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.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, 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.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional Tests for Hate Speech Detection Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.