Typologically Informed Parameter Aggregation

Stef Accou[◇†] and **Wessel Poelman**[◇]
[◇]LAGOM·NLP, Department of Computer Science, KU Leuven
[†]Department of Linguistics, KU Leuven
firstname.lastname@kuleuven.be

Abstract

Massively multilingual language models enable cross-lingual generalization but underperform on low-resource and unseen languages. While adapter-based fine-tuning offers a parameter-efficient solution, training language-specific adapters at scale remains costly. We introduce *Typologically Informed Parameter Aggregation* (TIPA), a training-free method that constructs proxy language adapters by aggregating existing ones, weighted by typological similarity. Integrated into the MAD-X framework, these proxies enable zero-shot cross-lingual transfer without additional training. We evaluate TIPA on five NLP tasks and over 230 languages. TIPA consistently outperforms or matches baselines such as English-only fine-tuning or selecting the typologically closest language adapter. We see the largest gains for languages lacking dedicated adapters. Our results demonstrate that typologically informed aggregation provides a viable alternative to language-specific modules without any training needed.[1]

1 Introduction

Massively multilingual language models have substantially expanded the reach of natural language processing (NLP), enabling zero-shot transfer across hundreds of languages. However, their performance remains uneven: while high-resource languages benefit from strong representations, models for under-resourced languages consistently underperform. This imbalance limits the inclusivity of NLP technologies and restricts their applicability for the majority of the world's languages (Joshi et al., 2020).

Despite rapid advances, the development of state-of-the-art methods for cross-lingual transfer has so far been restricted to a relatively narrow set of languages. This limits claims about their generalizability, as they lack evidence from a typologically diverse sample. Cross-lingual transfer is especially relevant in low-resource settings, where adapting existing models to new languages can benefit from knowledge transferred from better-resourced ones. Parameter-efficient fine-tuning (PEFT) approaches, particularly adapters (Houlsby et al., 2019), provide a practical solution to adapting existing models. Adapter modules are lightweight, reusable components that can be inserted into pre-trained transformer models. In multilingual settings, language adapters have proven effective by improving performance with minimal training required (*e.g.,* Pfeiffer et al., 2020b; Üstün et al., 2022; Klimaszewski et al., 2025).

While adapters are more efficient than full fine-tuning, they still require separate training for each language. Extending coverage to a broad set of languages remains difficult, especially since annotated data or monolingual corpora are scarce for certain languages. What if we could instead re-use existing adapters and construct new ones for a new language? This would be more scalable, foregoing the need for language-specific training.

Research Question. We investigate whether linguistic information can be used to guide cross-lingual transfer without training language-specific adapter modules. Specifically, we ask:

> *Can we use typological information to construct proxy language adapters via parameter aggregation?*

Our contributions are: (1) We introduce a typologically weighted aggregation method for constructing proxy language adapters in a training-free setting. By recombining existing adapters according to typology-based language proximity, we obtain stand-in modules that can be used in the MAD-X framework (Pfeiffer et al., 2020b). (2) We evaluate the method and a selection of strong baselines on a sample of 234 languages over five downstream

[1]Our code is available at: github.com/stefaccou/TIPA.

Findings of the Association for Computational Linguistics: EACL 2026, pages 2253–2267

tasks. Our results highlight the scalability of the introduced parameter aggregation method and show that training-free combination of modules can work when using typological similarity as a prior.

2 Related Work

Cross-Lingual Transfer and PEFT. Multilingual models can be adapted to new languages. Two popular ways of doing this is through continued pretraining, where the full weights of the model are updated, or through PEFT, where a small number of (additional) weights are updated. Both approaches have been used to enable cross-lingual transfer. Adapters have been successful in this area since they reduce computational costs while often preserving or increasing performance (Houlsby et al., 2019; Üstün et al., 2020, 2022). The MAD-X framework (Pfeiffer et al., 2020b,a; Poth et al., 2023) is particularly effective for cross-lingual transfer, offering modularity and composability across languages and tasks. While MAD-X-based approaches demonstrate the promise of modularity, they assume access to target-specific adapters, which can be costly to train.

Aggregation and Fusion Strategies. To extend coverage without per-language training, aggregation methods exploit the modular nature of adapters. Representation-level techniques such as AdapterFusion (Pfeiffer et al., 2021), EMEA (Wang et al., 2021), and more recently FLARE (Borchert et al., 2025) guide transfer by combining outputs from multiple adapters, at the cost of additional training and higher inference overhead. Parameter-level methods instead merge adapter weights directly, including AdapterSoup (Chronopoulou et al., 2023), task arithmetic (Ilharco et al., 2023), and recent language arithmetic approaches (Chronopoulou et al., 2024; Klimaszewski et al., 2025). These are training-free and inference-efficient but typically limited to pairwise or small-scale settings, often relying on carefully selected adapter combinations.

Typologically Guided Transfer. Linguistic typology is the study of structural commonalities and differences between languages. Typological annotations can be collected in databases, which is the main way this information is used in NLP (Littell et al., 2017; Baylor et al., 2023; Haynie et al., 2023; Khan et al., 2025). These databases provide a feature vector per language, which allows for calculating distances between languages (Ploeger et al., 2024, 2025). Since adapters also operate on the level of languages, typological feature vectors are a logical candidate for informing cross-lingual transfer. This has been used in adapter generation (Üstün et al., 2020; Ansell et al., 2021; Üstün et al., 2022) and in ensemble methods such as entropy-minimized aggregation (Wang et al., 2021), showing that typological priors can be effective, particularly for low-resource languages.

Existing research shows the promise of modularity, parameter aggregation, and typologically-informed transfer. To our knowledge, no prior work combines these strands to construct training-free, proxy language adapters. Our method introduces a typologically weighted aggregation algorithm that leverages existing adapters to approximate language-specific modules without retraining.

3 Methodology

We build on the MAD-X framework (Pfeiffer et al., 2020b). In this setting, a frozen multilingual transformer model is equipped with (1) a task adapter, trained on labeled data in a source language (typically English), and (2) a language adapter, trained on a language modeling task on a monolingual corpus. During training, the task adapter is stacked on top of the source language adapter. At inference time, the source language adapter can be substituted by a target language adapter, while keeping the task adapter fixed. This modularity enables separation of task and language knowledge.[2]

However, dedicated target-language adapters are not always available, especially for under-resourced languages. To address this, we introduce *Typologically Informed Parameter Aggregation* (TIPA), a method for constructing stand-in language adapters by aggregating parameters from existing adapters. TIPA requires no additional training and can be applied to any language for which typological information is available. The resulting proxy adapter can then be integrated into the MAD-X architecture.

Adapter Resources. We assume access to a pool of pretrained language adapters, which serve as building blocks for the construction of the proxy adapter, and to task adapters trained on English task data. The only requirement is that the languages represented in the pool are also represented in the typological database used.

[2]Although it is disputed whether this actually works as well as reported (Kunz and Holmström, 2024).

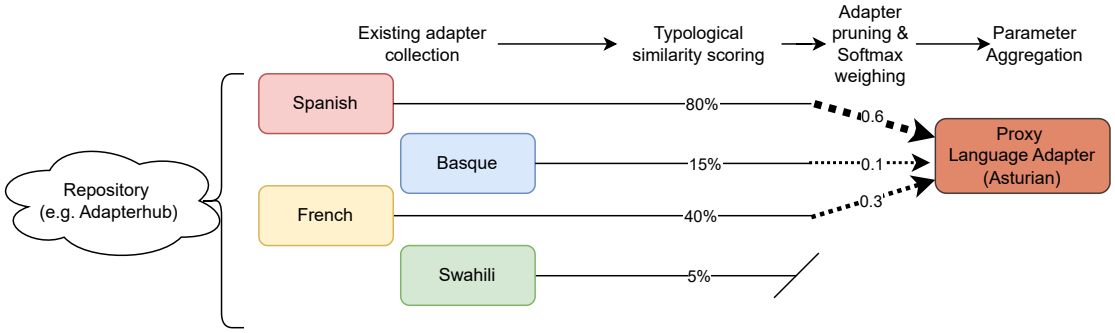

Figure 1 – Typologically Informed Parameter Aggregation (TIPA): a training-free framework that constructs a proxy language adapter for a target language by aggregating existing ones, weighted by typological similarity.

Typological Weighting. To approximate an adapter for target language ℓ_{tgt}, TIPA computes the proximity to the target for all source languages in the sample of available language adapters $\ell_{\text{src}} \in \mathcal{L}$. Distances are derived from the URIEL+ database (Khan et al., 2025), which encodes a wide range of linguistic dimensions, including morphology, syntax, phonology, and character inventories. For each target language, pairwise distances (d) are normalized and converted into similarity weights through a softmax function:

$$w_s = \frac{\exp(1 - d(\ell_{\text{tgt}}, \ell_{\text{src}}))}{\sum_{\ell'_{\text{src}} \in \mathcal{L}} \exp(1 - d(\ell_{\text{tgt}}, \ell'_{\text{src}}))}$$

Parameter Aggregation. Given the weighted set of source languages, TIPA constructs a new proxy adapter L_{proxy} through layer-wise parameter aggregation. Corresponding weight and bias matrices across all contributing language adapters are combined into a single module using the typologically deduced weights. The resulting proxy adapter matches the architecture of its source components and can thus be inserted directly into the MAD-X pipeline with the base model.

Example. Table 1 shows some example proxy adapters for a given target language. These are constructed using the adapters listed in Table 6.

Target language	Proxy created from (Top-3)		
Afrikaans	German (0.215)	Estonian (0.206)	Icelandic (0.195)
Tibetan	Min Dong (0.204)	Javanese (0.204)	Japanese (0.194)
Catalan	Italian (0.215)	Spanish (0.199)	Greek (0.196)

Table 1 – Top languages picked by TIPA to combine into a proxy adapter for the given target language.

4 Evaluation

We evaluate TIPA using XLM-RoBERTa base (Conneau et al., 2020) across five standard benchmarks: NER, POS tagging, COPA, QA, and topic classification, covering 234 languages. Detailed task descriptions, adapter resource configurations, and full language breakdowns are provided in §A.1. We compare against the following baselines:

- **English-only fine-tuning:** Fine-tune the model on English task data, then apply zero-shot to the target language.
- **MAD-X with target-language adapter:** If available, a dedicated language adapter for the target language is used during inference. If no language adapter is available, we use the adapter trained on the typologically closest language.
- **No Train but Gain:** A re-implementation of (Klimaszewski et al., 2025): additive combination of the English adapter with the typologically most similar available adapter.
- **Uniform averaging:** Proxy adapters created by unweighted parameter averaging across all available language adapters in the sample.

5 Results

	ALL	NER	POS	COPA	QA	SIB
# Languages	*(234)*	*(136)*	*(80)*	*(11)*	*(12)*	*(176)*
Fine-tuning	45.0 ± 20.7	39.0 ± 18.4	38.9 ± 18.8	**55.5 ± 3.0**	53.4 ± 14.9	61.2 ± 25.0
MAD-X	45.4 ± 21.1	43.3 ± 21.4	44.6 ± 20.9	52.0 ± 2.1	72.1 ± 5.0	56.5 ± 24.3
No Train but Gain	50.1 ± 21.4	49.3 ± 18.1	**46.9 ± 20.5**	50.3 ± 1.4	72.5 ± 5.4	61.6 ± 25.1
Parameter averaging	43.1 ± 20.4	39.5 ± 18.1	45.4 ± 20.8	51.5 ± 2.0	68.6 ± 6.1	54.3 ± 25.1
TIPA	**54.1 ± 22.9**	**51.3 ± 18.5**	46.8 ± 20.7	51.8 ± 2.3	**72.9 ± 5.2**	**63.4 ± 24.3**

Table 2 – Baseline and TIPA scores aggregated across all tasks, alongside task-specific results. Scores are reported ×100. The color scheme situates relative performance per task. Full results for each task are in §A.3.

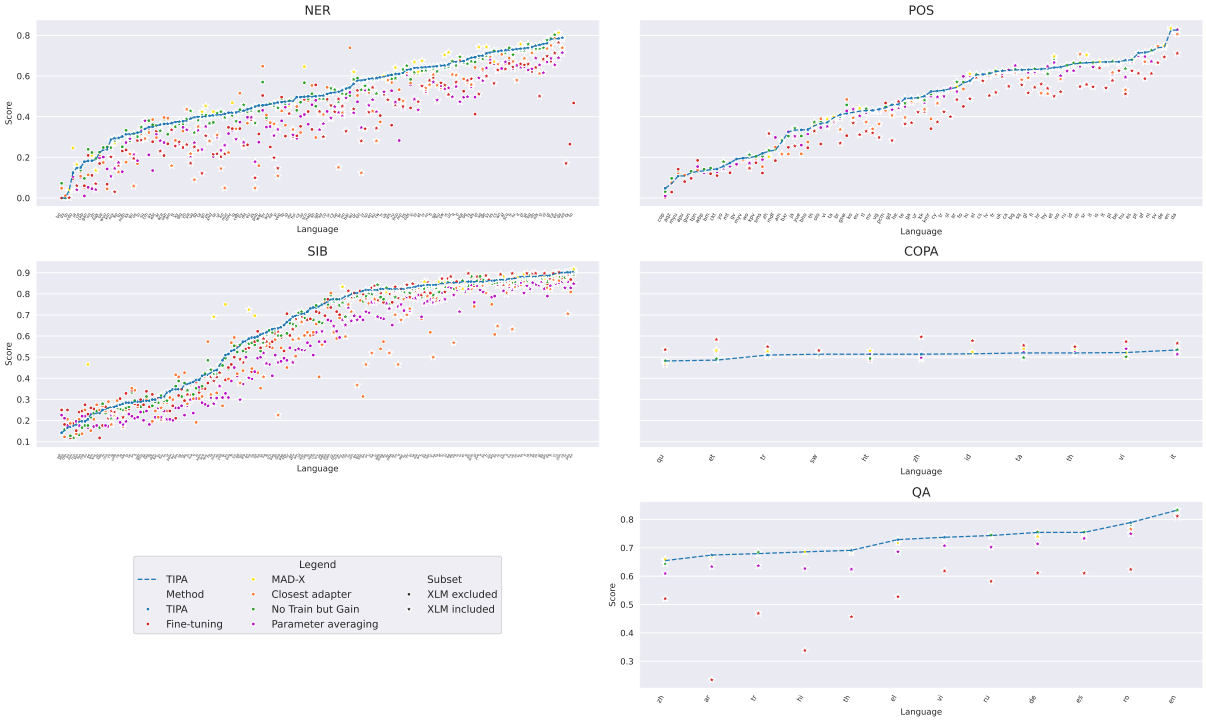

Figure 2 – Scores for all tasks across available languages. We compare TIPA to *No Train but Gain*, fine-tuning, uniform parameter averaging, and to either MAD-X or the typologically closest adapter, depending on adapter availability. Languages are marked for inclusion in XLM-R, and ordered based on increasing performance of our method. Larger formats of each presented figure can be found in §A.2, alongside a more detailed analysis.

Overall Results. As shown in Table 2, TIPA outperforms all the evaluated baselines when aggregating results over all tasks.[3] The improvements are statistically significant under paired one–tailed t-tests (all comparisons $p < 0.01$). Improvements are particularly pronounced for languages that lack a dedicated language adapter, where the proxy module directly fulfills its intended role as a stand-in within MAD-X. A more detailed breakdown by resource setting is presented in §A.2. TIPA yields consistent gains relative to the widely used baseline of English-only fine-tuning, indicating that typology-aware parameter aggregation provides a useful prior compared to relying solely on the generalization capacity of the underlying model. TIPA improves over unweighted parameter averaging by +6.7% on average (task-aggregated; $p < 0.01$), confirming the value of linguistically informed weighting. Importantly, these improvements are achieved without any additional language-specific fine-tuning or adapter training.

Task-level Results. As shown in Figure 2, the benefits of TIPA are largest on morphologically sensitive tasks such as NER and POS. For higher-level semantic tasks (COPA, QA, SIB), TIPA remains competitive with the strongest adapter-based baselines and, in several settings, matches or surpasses the performance of dedicated pre-trained language adapters. Our re-implementation of *No Train but Gain* (Klimaszewski et al., 2025), with automatic selection of the second adapter is consistently equalled or outperformed by TIPA, although it remains a strong second-best baseline. This confirms that, while other parameter aggregation methods can be competitive, aggregating information from multiple sources yields more robust gains than combining English with a single, typologically closest adapter. Together, these results indicate that typologically-weighted aggregation is a viable, training-free alternative to per-language modules, with the strongest benefits in token-level tasks (Figure 2) and where dedicated adapters are unavailable. Nevertheless, performance of all methods varies greatly across tasks. A more detailed overview is provided in §A.2.

[3] Aggregating results like this has been rightfully criticized (Pikuliak and Simko, 2022; Ploeger et al., 2024), we do it here to show the viability of the method. In the following sections we provide more detailed analyses.

Feature Type Ablation and Pruning. The main parameters in TIPA specify which typological features are included in the distance calculation and how many adapters are considered in creating the proxy adapter.

We investigate which feature categories from URIEL+ are the most effective. We specifically consider: *featural* (all available features), *morphological*, and *syntactic* for the language feature vectors. We find that for NER and POS, distances based on the *morphological* category emerge as the most informative, while *featural* remains the best overall choice. The full results of this can be found in Table 7.

We also apply pruning strategies that either (1) retain the top-k most similar source adapters for aggregation, instead of all available adapters or (2) only include adapters with a certain distance score to the target language based on a threshold. We find there are some gains to be had from either pruning approach, but see no clear winner. Investigating more detailed refining of distance metrics and pruning methods is left for future work. Full results and further explanations are provided in §A.3.

6 Conclusion

We introduce TIPA, a training-free and parameter-efficient method for extending multilingual models to languages without dedicated adapters. By aggregating existing language adapters according to typological similarity, our method constructs proxy modules that integrate into the MAD-X framework. Evaluations across five tasks and more than 230 languages show improvements over strong baselines, with the largest gains for languages lacking adapters or excluded from pretraining.

Beyond gains on tasks, TIPA shows the benefit of using linguistic knowledge as an inexpensive but effective prior for cross-lingual transfer.

Typology-informed aggregation provides a practical way to improve cross-lingual transfer for underrepresented languages without increasing training or inference costs. In our experiments, this approach consistently benefited languages without model and adapter coverage. Future work should explore extending TIPA to multilingual models beyond XLM-R and investigating its use with task adapters and multi-task or full-parameter fine-tuning.

Limitations

Our findings have several limitations. First, performance varies substantially across tasks. While TIPA yields gains even for languages not included in pretraining, the performance of the underlying multilingual model deteriorates sharply for unseen scripts or unseen tokens, a limitation that our method cannot address due to the lack of meaningful representations.

Second, a key assumption of our approach is the availability of a set of language adapters for a given model. The effectiveness depends on the quality, breadth, and diversity of the pool of available adapters. We did not explore these aspects in detail in the present study.

Third, although factors such as feature vector selection, similarity thresholds, or adapter selection can yield improvements, these factors were tuned heuristically. We did not perform an exhaustive search over these parameters.

Fourth, despite the inclusion of 234 languages (depending on the task), our study remains subject to bibliographic bias: sufficient corpora and evaluation datasets exist for only a fraction of the world's languages (Rijkhoff and Bakker, 1998). As a result, the reported gains for "low-resource" languages are only for those that have enough available resources for us to evaluate.

Finally, preliminary experiments on conditional multilingual language models (Gemma, Qwen) and PEFT methods (LoRa) did not see strong results. This suggests that the effectiveness of typology-aware aggregation may be architecture-dependent, and that further validation is needed across a broader range of model families.

Acknowledgments

This paper builds on the results from the first author's Master Thesis, completed as part of the Master of Artificial Intellence programme at KU Leuven (Accou, 2025). The authors want to thank Miryam de Lhoneux for her supervision and guidance throughout the research process. We also thank the anonymous reviewers for their constructive feedback. WP is funded by a KU Leuven Bijzonder Onderzoeksfonds C1 project with reference C14/23/096. The computational resources and services used were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI.

References

Stef Accou. 2025. TIPA: Typologically Informed Parameter Aggregation. Master's thesis, KU Leuven. Faculteit Ingenieurswetenschappen.

David Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba Alabi, Yanke Mao, Haonan Gao, and En-Shiun Lee. 2024. SIB-200: A Simple, Inclusive, and Big Evaluation Dataset for Topic Classification in 200+ Languages and Dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual Adapter Generation for Efficient Cross-Lingual Transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Emi Baylor, Esther Ploeger, and Johannes Bjerva. 2023. The Past, Present, and Future of Typological Databases in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1163–1169. Association for Computational Linguistics.

Philipp Borchert, Ivan Vulić, Marie-Francine Moens, and Jochen De Weerdt. 2025. Language Fusion for Parameter-Efficient Cross-lingual Transfer. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25848–25868, Vienna, Austria. Association for Computational Linguistics.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexandra Chronopoulou, Jonas Pfeiffer, Joshua Maynez, Xinyi Wang, Sebastian Ruder, and Priyanka Agrawal. 2024. Language and Task Arithmetic with Parameter-Efficient Layers for Zero-Shot Summarization. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 114–126. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Hannah J. Haynie, Damián Blasi, Hedvig Skirgård, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023. Grambank's Typological Advances Support Computational Research on Diverse Languages. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 147–149. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Aditya Khan, Mason Shipton, David Anugraha, Kaiyao Duan, Phuong H. Hoang, Eric Khiu, A. Seza Doğruöz, and En-Shiun Annie Lee. 2025. URIEL+: Enhancing Linguistic Inclusion and Usability in a Typological and Multilingual Knowledge Base. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6937–6952. Association for Computational Linguistics.

Mateusz Klimaszewski, Piotr Andruszkiewicz, and Alexandra Birch. 2025. No Train but Gain: Language Arithmetic for training-free Language Adapters enhancement. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11121–11134. Association for Computational Linguistics.

Jenny Kunz and Oskar Holmström. 2024. The Impact of Language Adapters in Cross-Lingual Transfer for NLU. In *Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024)*, pages 24–43. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological,

geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, Daniel Zeman, and 1 others. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043. European Language Resources Association.

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. 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. 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.

Matúš Pikuliak and Marian Simko. 2022. Average Is Not Enough: Caveats of Multilingual Evaluation. In *Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL)*, pages 125–133. Association for Computational Linguistics.

Esther Ploeger, Wessel Poelman, Miryam de Lhoneux, and Johannes Bjerva. 2024. What is "Typological Diversity" in NLP? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5681–5700. Association for Computational Linguistics.

Esther Ploeger, Wessel Poelman, Andreas Holck Høeg-Petersen, Anders Schlichtkrull, Miryam de Lhoneux, and Johannes Bjerva. 2025. A Principled Framework for Evaluating on Typologically Diverse Languages. *Computational Linguistics*, pages 1–36.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A Unified Library for Parameter-Efficient and Modular Transfer Learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively Multilingual Transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Jan Rijkhoff and Dik Bakker. 1998. Language sampling. *Linguistic Typology*, 2(3):263–314.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. *AAAI Spring Symposium Series*.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language Adaptation for Truly Universal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2022. UDapter: Typology-based Language Adapters for Multilingual Dependency Parsing and Sequence Labeling. *Computational Linguistics*, 48(3):555–592.

Xinyi Wang, Yulia Tsvetkov, Sebastian Ruder, and Graham Neubig. 2021. Efficient Test Time Adapter Ensembling for Low-resource Language Varieties. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 730–737, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Tasks and Resource Settings

To ensure representativeness, we evaluate across five standard NLP benchmarks spanning multiple linguistic levels:

- NER (WikiAnn; Rahimi et al., 2019), a token-level classification task centered on identifying named entities.
- POS tagging (Universal Dependencies; Nivre et al., 2020), a sequence-level annotation task assigning part-of-speech labels to words.
- COPA (XCOPA; Roemmele et al., 2011; Ponti et al., 2020), a sentence-level reasoning task, requiring causal commonsense inference.
- QA (XQuAD; Rajpurkar et al., 2016; Artetxe et al., 2020), a span-based question answering task.
- Topic Classification (SIB-200; Adelani et al., 2024), a simple document-level topic classification task across a broad sample of over 200 languages and dialects.

Together these tasks cover token-level, sequence-level, and document-level evaluation. Our sample comprises 234 languages, including high-resource languages from XLM-R pre-training, languages with existing MAD-X adapters, and languages without dedicated adapters that thus require proxy reconstruction. NER, POS and SIB span the most typologically diverse samples of languages, serving as robust benchmarks for assessing the method's generalisability across languages with diverse resource scenarios. Table 3 summarises the number of languages per task and their distribution across these resource categories, and Tables 4 and 5 give a full breakdown of languages included in each task.

For each target language, we construct a proxy adapter using TIPA and combine it with the relevant task adapter under the MAD-X framework. Performance is measured on the test sets of each task. All models are evaluated without additional training, ensuring that results reflect the zero-shot generalisation capacity of proxy adapters.

A.2 Task-level Results

Figure 3compares TIPA to the baseline methods for all tasks and languages. The benefits of TIPA are largest on morphology-sensitive tasks such as NER and POS, where typological proximity provides informative guidance for reconstructing language-specific behaviour. In these cases, the improvements can be further augmented by limiting the similarity calculation process to only include morphological information. For NER, TIPA yields substantial gains, with particularly strong improvements for languages without a dedicated adapter, where it statistically outperforms all baselines ($p < 0.01$). In contrast, for languages with an existing adapter, performance is slightly below the MAD-X baseline. Further analysis shows that while higher token overlap consistently benefits all methods, TIPA achieves larger relative improvements in low-overlap settings. For POS, improvements are more modest and concentrated in languages natively supported by the underlying model.

For higher-level semantic tasks (COPA, QA, SIB), TIPA remains competitive with the strongest adapter-based baselines and, in most settings, matches or surpasses the performance of dedicated pre-trained language adapters. In COPA, variance across methods is smaller. Unlike in other tasks, fine-tuning achieves the strongest results for COPA, possibly due to the fact that all languages in the test set are natively supported high-resource languages. For QA, TIPA statistically outperforms all methods ($p \leq 0.05$), except for the No Train but Gain baseline ($p = 0.10$), and even consistently outperforms the MAD-X baseline when a dedicated adapter is available. This indicates that for this reasoning task, updating the dedicated adapter with parameters from typologically related languages provides additional benefit over monolingual adapters. Finally, for SIB, TIPA achieves the strongest overall results ($p < 0.001$), significantly outperforming all baselines across conditions. The only exception is the subset of languages with a dedicated adapter, where performance falls short of the monolingual MAD-X adapter baseline.

A.3 Distance Type and Language Pruning

All prior experiments employed the *featural distance* metric from URIEL+ (Khan et al., 2025) to calculate the typological similarity scores, which serve as the basis for weighting source language adapters in the approximation of target-language adapters. The featural distance metric is a composite of syntactic, morphological, phonological, and genetic information, offering the broadest coverage among available distance types. Also, it should be noted that in previous experiments, the full set of

Task	Total languages	With adapter	Without adapter	In XLM-R	Not in XLM-R
NER	134	29	105	85	49
POS	80	19	61	57	23
COPA	11	11	0	11	0
QA	12	11	1	12	0
SIB	176	25	151	81	95
Total	**234**	**31**	**203**	**95**	**139**

Table 3 – Evaluated languages per task, along with resource-specific subsets.

source adapters is weighted and integrated in the composite proxy adapter, regardless of their individual score. While more distant adapters receive a lower weight, they still contribute to the final proxy adapter.

To explore whether further gains in performance can be achieved within the TIPA method, we introduce two variations to the aggregation process:

1. **Varying the distance type used for calculating adapter weights**
   We evaluate if the use of morphological or syntactic distance has an impact on the score of the resulting reconstructed adapter. It should be noted that these distance types have narrower language coverage than the combined *featural* metric. As a result, the number of evaluated languages is reduced from 234 to 207, with lower-resourced languages being disproportionately affected. The remaining distance types (such as phonological or genetic distance) provide even lower coverage, and are thus omitted from this analysis, as their inclusion would conflict with our goal of maintaining a maximally linguistically broad and inclusive evaluation setup.

2. **Restricting the number of source adapters used in the reconstruction process**
   We evaluate two strategies:

   (a) A fixed limit, where only the top five source languages (based on proximity for the different distance types) are retained during the softmax-weighted aggregation.

   (b) A distance threshold, where only languages with a similarity score above 0.33 are considered and adequately weighted for inclusion in the reconstruction.

Distance Types. For the NER and POS tagging tasks, morphological distance emerges as the most informative typological distance. In these tasks, it is the only distance type for which TIPA consistently outperforms all baselines with statistical significance ($p \leq 0.05$), including the *No Train but Gain* baseline, which could not be reliably surpassed under the default *featural distance* setting. In contrast, syntactic distance proves less effective, yielding comparatively weaker results. For both methods, morphological distance just falls short of outperforming featural distance with statistical significance. Nevertheless, we consider morphological distance to be particularly relevant for tasks that rely more heavily on morphological cues, such as NER and POS.

For the COPA and QA tasks, the impact of distance type is minimal. Performance differences between morphological and syntactic distances remain within a margin of 0.2%, rendering the variation negligible. Consequently, the results mirror those discussed in subsection A.2, with no change in relative performance against the evaluated baselines. For SIB, which is evaluated on a larger language sample, syntactic distance emerges as the most informative similarity criterium. It statistically outperforms all other distance types ($p \leq 0.01$). Using syntactic distance in the approximation function results in a 0.5% performance gain relative to featural distance on average. In contrast, morphological distance consistently underperforms. This can be due to the fact that morphological clues are more informative in surface-level tasks such as NER and POS, while syntactic similarity results in better transfer for higher-level tasks.

Limitation Strategies. Limitation strategies have a stronger impact on TIPA scores than distance type variations alone. Aggregating over all tasks, both the fixed-limit and the threshold-based strategies significantly outperform their corresponding unmodified variants ($p \leq 0.03$). In particular, limiting the number of source adapters

appears to be most effective when using the *syntactic* distance metric. For all tasks, the best overall results achieved by our method are consistently associated with one of these limitation strategies, as shown in Table 7.

We leave the exploration of limit and threshold values for further research. The results presented here indicate that refining these parameters on a per-task basis may lead to further improvements. This suggests that our TIPA approach remains open to straightforward optimisations, and that even minor adjustments could yield additional gains beyond the current implementation in cross-lingual transfer settings.

Language	NER	COPA	POS	QA	SIB
Achinese (ace)	1	0	0	0	1
Afrikaans (af)	1	0	1	0	1
Albanian (sq)	1	0	1	0	0
Amharic (am)	1	0	1	0	1
Arabic (ar)	1	0	1	1	0
Armenian (hy)	1	0	1	0	1
Assamese (as)	1	0	0	0	1
Bambara (bm)	0	0	1	0	1
Bashkir (ba)	1	0	0	0	1
Basque (eu)	1	0	1	0	1
Belarusian (be)	1	0	1	0	1
Bengali (bn)	1	0	0	0	1
Bhojpuri (bho)	0	0	1	0	1
Bosnian (bs)	1	0	0	0	1
Breton (br)	1	0	1	0	0
Bulgarian (bg)	1	0	1	0	1
Catalan (ca)	1	0	1	0	1
Cebuano (ceb)	1	0	0	0	1
Chinese (zh)	1	1	1	1	1
Crimean Tatar (crh)	1	0	0	0	1
Croatian (hr)	1	0	1	0	1
Czech (cs)	1	0	1	0	1
Danish (da)	1	0	1	0	1
Dutch (nl)	1	0	1	0	1
Egyptian Arabic (arz)	1	0	0	0	1
English (en)	1	0	1	1	1
Esperanto (eo)	1	0	0	0	1
Estonian (et)	1	1	1	0	1
Faroese (fo)	1	0	1	0	1
Finnish (fi)	1	0	1	0	1
French (fr)	1	0	1	0	1
Friulian (fur)	1	0	0	0	1
Galician (gl)	1	0	1	0	1
Georgian (ka)	1	0	0	0	1
German (de)	1	0	1	1	1
Greek (el)	1	0	1	1	1
Guarani (gn)	1	0	0	0	1
Gujarati (gu)	1	0	0	0	1
Haitian Creole (ht)	0	1	0	0	1
Hebrew (he)	1	0	1	0	1
Hindi (hi)	1	0	1	1	1
Hungarian (hu)	1	0	1	0	1
Icelandic (is)	1	0	1	0	1
Igbo (ig)	1	0	0	0	1
Ilocano (ilo)	1	0	0	0	1
Indonesian (id)	1	1	1	0	1
Irish (ga)	1	0	1	0	1
Italian (it)	1	1	1	0	1
Japanese (ja)	1	0	1	0	1
Javanese (jv)	1	0	0	0	1
Kannada (kn)	1	0	0	0	1
Kazakh (kk)	1	0	1	0	1
Khmer (km)	1	0	0	0	1
Kinyarwanda (rw)	1	0	0	0	1

Language	NER	COPA	POS	QA	SIB
Korean (ko)	1	0	1	0	1
Kyrgyz (ky)	1	0	0	0	1
Latvian (lv)	1	0	1	0	0
Lingala (ln)	1	0	0	0	1
Lithuanian (lt)	1	0	1	0	1
Lombard (lmo)	1	0	0	0	1
Luxembourgish (lb)	1	0	0	0	1
Macedonian (mk)	1	0	0	0	1
Malayalam (ml)	1	0	0	0	1
Maltese (mt)	1	0	1	0	1
Maori (mi)	1	0	0	0	1
Marathi (mr)	1	0	1	0	1
Minangkabau (min)	1	0	0	0	1
Myanmar (Burmese) (my)	1	0	0	0	1
Northern Kurdish (kmr)	0	0	1	0	1
Norwegian (no)	1	0	1	0	0
Occitan (oc)	1	0	0	0	1
Polish (pl)	1	0	1	0	1
Portuguese (pt)	1	0	1	0	1
Punjabi (pa)	1	0	0	0	1
Quechua (qu)	1	1	0	0	0
Romanian (ro)	1	0	1	1	1
Russian (ru)	1	0	1	1	1
Scots Gaelic (gd)	1	0	1	0	1
Serbian (sr)	1	0	1	0	1
Sicilian (scn)	1	0	0	0	1
Sindhi (sd)	1	0	0	0	1
Sinhala (si)	1	0	0	0	1
Slovenian (sl)	1	0	1	0	1
Somali (so)	1	0	0	0	1
Sorani Kurdish (ckb)	1	0	0	0	1
Spanish (es)	1	0	1	1	1
Sundanese (su)	1	0	0	0	1
Swahili (sw)	1	1	0	0	0
Swedish (sv)	1	0	1	0	1
Tagalog (tl)	1	0	1	0	1
Tajik (tg)	1	0	0	0	1
Tamil (ta)	1	1	1	0	1
Tatar (tt)	1	0	0	0	1
Telugu (te)	1	0	1	0	1
Thai (th)	1	1	1	1	1
Tibetan (bo)	1	0	0	0	1
Tosk Albanian (als)	1	0	0	0	1
Turkish (tr)	1	1	1	1	1
Turkmen (tk)	1	0	0	0	1
Ukrainian (uk)	1	0	1	0	1
Urdu (ur)	1	0	1	0	1
Uyghur (ug)	1	0	1	0	1
Vietnamese (vi)	1	1	1	1	1
Waray (war)	1	0	0	0	1
Welsh (cy)	1	0	1	0	1
Wolof (wo)	0	0	1	0	1
Yoruba (yo)	1	0	1	0	1
Yue Chinese (yue)	0	0	1	0	1

Table 4 – Language coverage per task.

Task	Languages
COPA	Akuntsu (aqz), Apurinã (apu), Chukot (ckt), Coptic (cop), Erzya (myv) Komi-Zyrian (kpv), Livvi (olo), Manx (gv), Mbyá Guaraní (gun), Moksha (mdf) Mundurukú (myu), Nigerian Pidgin (pcm), Russia Buriat (bxr), Skolt Sami (sms) Swiss German (gsw), Tupinambá (tpn), Warlpiri (wbp)
NER	Aragonese (an), Aymara (ay), Bavarian (bar), Chechen (ce), Chuvash (cv) Corsican (co), Dhivehi (dv), Dimli (diq), Eastern Mari (mhr), Extremaduran (ext) Frisian (fy), Gan Chinese (gan), Hakka Chinese (hak), Kurmanji Kurdish (ku) Low German (nds), Mazanderani (mzn), Min Dong Chinese (cdo), Mingrelian (xmf) Mongolian (mn), Neapolitan (nap), Nepali (ne), Northern Frisian (frr) Ossetian (os), Pashto (ps), Romansh (rm), Scots (sco), Serbo-Croatian (sh) Uzbek (uz), Veps (vep), Vlaams (vls), Western Panjabi (pnb), Wu Chinese (wuu) Yakut (sah), Zeeuws (zea)
POS	/
QA	/
SIB	Awadhi (awa), Ayacucho Quechua (quy), Balinese (ban), Bemba (bem) Buginese (bug), Central Atlas Tamazight (tzm), Central Aymara (ayr) Central Kanuri (knc), Chichewa (ny), Dari (prs), Dyula (dyu), Dzongkha (dz) Eastern Yiddish (ydd), Ewe (ee), Fijian (fj), Fon (fon), Halh Mongolian (khk) Hausa (ha), Iranian Persian (pes), Kabiyè (kbp), Kabuverdianu (kea) Kabyle (kab), Kachin (kac), Kamba (kam), Kashmiri (ks), Kikuyu (ki), Lao (lo) Luba-Lulua (lua), Luganda (lg), Luo (luo), Magahi (mag), Maithili (mai) Meiteilon (Manipuri) (mni), Mesopotamian Arabic (acm), Mizo (lus) Moroccan Arabic (ary), Mossi (mos), Najdi Arabic (ars), Nepali (npi) Nigerian Fulfulde (fuv), North Azerbaijani (azj), North Levantine Arabic (apc) Northern Uzbek (uzn), Norwegian Bokmål (nb), Nuer (nus), Odia (ory) Pangasinan (pag), Papiamento (pap), Plateau Malagasy (plt), Rundi (rn) Samoan (sm), Sango (sg), Santali (sat), Sepedi (nso), Sesotho (st), Shan (shn) Shona (sn), South Azerbaijani (azb), Southern Pashto (pbt) Standard Arabic (arb), Standard Malay (zsm), Swahili (swh), Swati (ss) Tamasheq (taq), Tigrinya (ti), Tok Pisin (tpi), Tsonga (ts), Tswana (tn) Tunisian Arabic (aeb), Twi (ak), Twi (tw), Umbundu (umb) West Central Oromo (gaz), Xhosa (xh), Zulu (zu)

Table 5 – Languages included in only one task.

Languages with existing adapters
Arabic, Basque, Chinese, Eastern Mari, English, Estonian, German, Greek, Guarani, Haitian Creole, Hindi, Icelandic, Ilocano, Indonesian, Italian, Japanese, Javanese, Maori, Min Dong Chinese, Mingrelian, Myanmar (Burmese), Quechua, Russian, Serbian, Spanish, Swahili, Tamil, Thai, Turkish, Turkmen, Vietnamese

Table 6 – Languages in our pool of existing MAD-X adapters trained for XLM-R. Retrieved from https://huggingface.co/collections/AdapterHub/mad-x-adapters

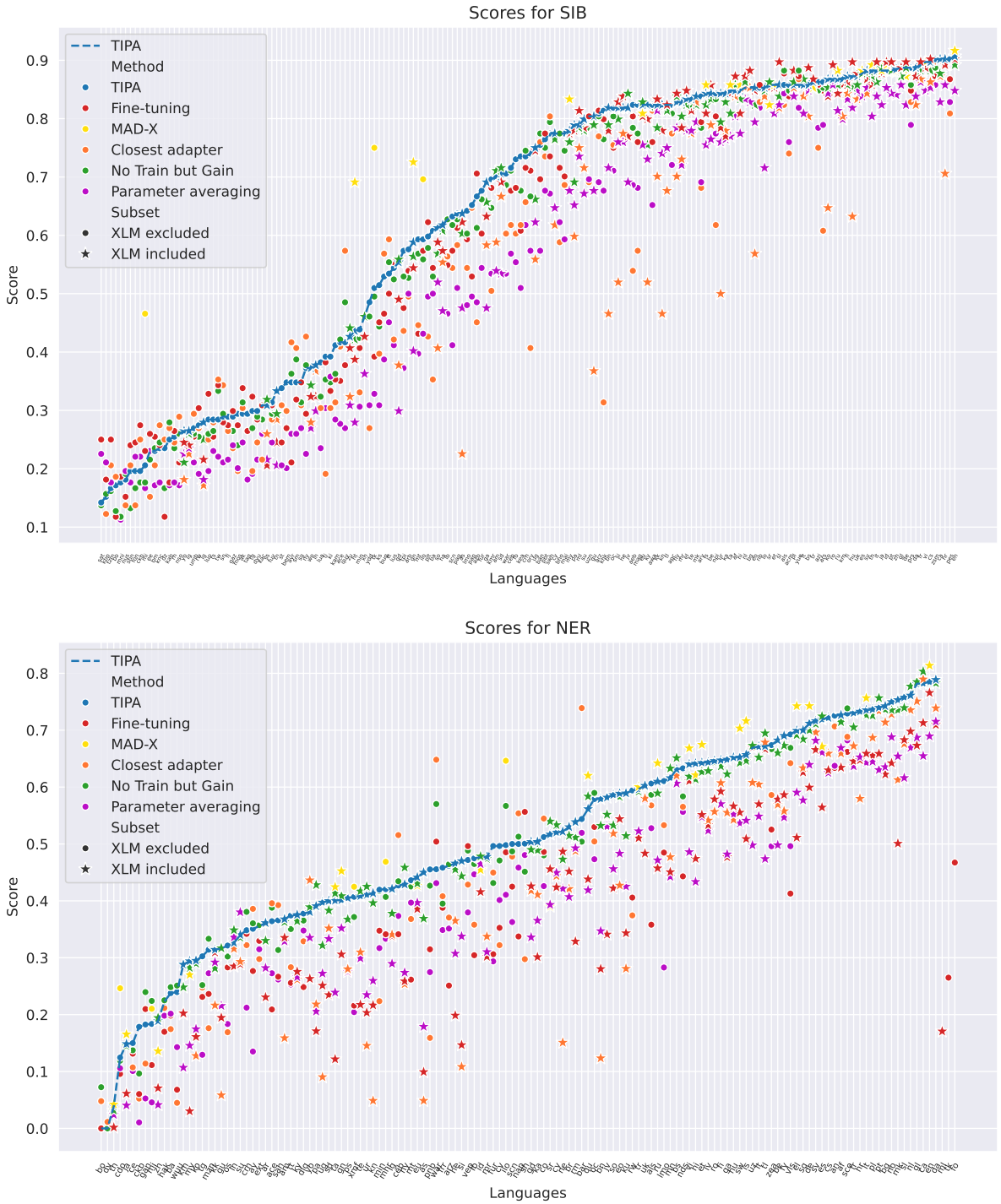

Figure 3 – Scores for each task, across all evaluated languages. Comparing TIPA to the *No Train but Gain* and fine-tuning baselines, as well as to either the MAD-X configuration or the typologically closest adapter, depending on adapter availability. Languages are presented in order of increasing performance of our method, and marked for native support in XLM-R. The results show that languages not included in XLM-R pre-training generally underperform, and that the benefits of our method primarily concern natively supported languages.

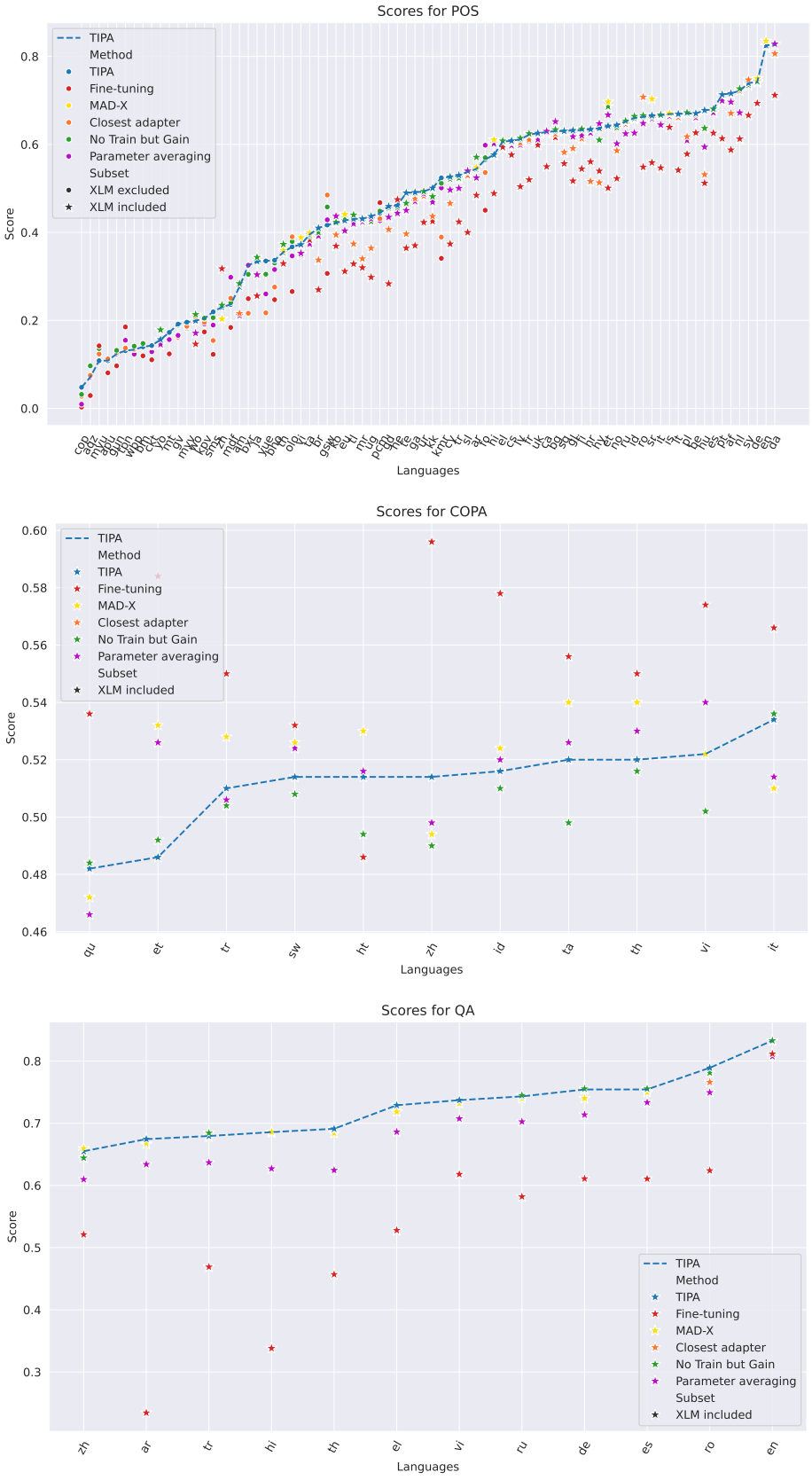

Figure 3 – Scores for each task, across all evaluated languages. Comparing TIPA to the *No Train but Gain* and fine-tuning baselines, as well as to either the MAD-X configuration or the typologically closest adapter, depending on adapter availability. Languages are presented in order of increasing performance of our method, and marked for native support in XLM-R. The results show that languages not included in XLM-R pre-training generally underperform, and that the benefits of our method primarily concern natively supported languages.

SUBSET	METHOD	ALL	NER	POS	COPA	QA	SIB
Overall	Fine-tuning	0.450	0.390	0.389	**0.555**	0.534	0.612
	MAD-X	0.454	0.433	0.446	0.520	0.721	0.565
	No Train but Gain	0.501	0.493	**0.469**	0.503	0.725	0.616
	TIPA$_{Featural+limit}$		**0.513**		0.518		
	TIPA$_{Featural}$			0.468			
	TIPA$_{Morphological+threshold}$					**0.729**	
	TIPA$_{Syntactic+limit}$	**0.541**					**0.634**
No adapter	Fine-tuning	0.447	0.395	0.367		0.624	0.585
	MAD-X	0.431	0.414	0.410		0.766	0.522
	No Train but Gain	0.491	0.496	0.439		0.781	0.590
	TIPA$_{Featural+limit}$	**0.520**	**0.516**				
	TIPA$_{Featural}$			**0.440**			
	TIPA$_{Morphological}$					**0.789**	
	TIPA$_{Syntactic+limit}$						**0.608**
With adapter	Fine-tuning	0.500	0.371	0.507	**0.555**	0.525	0.767
	MAD-X	0.605	0.505	0.561	0.520	0.717	**0.824**
	No Train but Gain	0.568	0.484	0.562	0.503	0.720	0.773
	TIPA$_{Featural+limit}$				0.518	**0.723**	
	TIPA$_{Morphological+limit}$						0.792
	TIPA$_{Morphological+threshold}$			**0.564**			
	TIPA$_{Syntactic+limit}$	**0.621**	**0.509**				
XLM included	Fine-tuning	0.530	0.425	0.485	**0.555**	0.534	0.754
	MAD-X	0.542	0.469	0.537	0.520	0.721	0.691
	No Train but Gain	0.605	0.542	0.560	0.503	0.725	0.754
	TIPA$_{Featural+limit}$		**0.562**		0.518		
	TIPA$_{Featural+threshold}$			**0.562**			
	TIPA$_{Featural}$						**0.768**
	TIPA$_{Morphological+threshold}$					**0.729**	
	TIPA$_{Syntactic}$	**0.653**					
XLM excluded	Fine-tuning	0.401	0.337	0.207			0.505
	MAD-X	0.393	0.371	0.221			0.458
	No Train but Gain	0.429	0.407	0.243			0.498
	TIPA$_{Morphological+threshold}$			**0.255**			
	TIPA$_{Syntactic+limit}$	**0.468**					**0.534**
	TIPA$_{Syntactic+threshold}$		**0.443**				

Table 7 – Baseline scores and best scores obtained for each subset and task. The highest score for each task within each subset of languages is put in bold, and the colour scheme situates the relative performance of the scores per task. The highest scoring method from our framework is presented, marked for distance type and limiting method, along with its score in the relevant task. The *ALL* column presents task-aggregated results, in which task scores are weighted based on the number of evaluated languages.