Meta-Learning for Fast Cross-Lingual Adaptation in Dependency Parsing

Anna Langedijk[★,1], **Verna Dankers**[1,2], **Phillip Lippe**[1], **Sander Bos**[1], **Bryan Cardenas Guevara**[1], **Helen Yannakoudakis**[3], **and Ekaterina Shutova**[1]

[1]Institute for Logic, Language and Computation, University of Amsterdam
[2]Institute of Language, Cognition and Computation, University of Edinburgh
[3]Department of Informatics, King's College London

Abstract

Meta-learning, or learning to learn, is a technique that can help to overcome resource scarcity in cross-lingual NLP problems, by enabling fast adaptation to new tasks. We apply model-agnostic meta-learning (MAML) to the task of cross-lingual dependency parsing. We train our model on a diverse set of languages to learn a parameter initialization that can adapt quickly to new languages. We find that meta-learning with pre-training can significantly improve upon the performance of language transfer and standard supervised learning baselines for a variety of unseen, typologically diverse, and low-resource languages, in a few-shot learning setup.

1 Introduction

The field of natural language processing (NLP) has seen substantial performance improvements due to large-scale language model pre-training (Devlin et al., 2019). Whilst providing an informed starting point for subsequent task-specific fine-tuning, such models still require large annotated training sets for the task at hand (Yogatama et al., 2019). This limits their applicability to a handful of languages for which such resources are available and leads to an imbalance in NLP technology's quality and availability across linguistic communities. Aiming to address this problem, recent research has focused on the development of multilingual sentence encoders, such as multilingual BERT (mBERT) (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), trained on as many as 93 languages. Such pre-trained multilingual encoders enable zero-shot transfer of task-specific models across languages (Wu and Dredze, 2019), offering a possible solution to resource scarcity. Zero-shot transfer, however, is most successful among typologically similar, high-resource languages, and less so for languages distant from the training languages and in resource-lean scenarios (Lauscher et al., 2020). This stresses the need to develop techniques for fast cross-lingual model adaptation, that can transfer knowledge across a wide range of typologically diverse languages with limited supervision.

In this paper, we focus on the task of universal dependency (UD) parsing and present a novel approach for effective and resource-lean cross-lingual parser adaptation via meta-learning, requiring only a small number of training examples per language (which are easy to obtain even for low-resource languages). Meta-learning is a learning paradigm that leverages previous experience from a set of tasks to solve a new task efficiently. As our goal is fast cross-lingual model adaptation, we focus on optimization-based meta-learning, where the main objective is to find a set of initial parameters from which rapid adaption to a variety of different tasks becomes possible (Hospedales et al., 2020). Optimization-based meta-learning has been successfully applied to a variety of NLP tasks. Notable examples include neural machine translation (Gu et al., 2018), semantic parsing (Huang et al., 2018), pre-training text representations (Lv et al., 2020), word sense disambiguation (Holla et al., 2020) and cross-lingual natural language inference and question answering (Nooralahzadeh et al., 2020). To the best of our knowledge, meta-learning has not yet been explored in the context of dependency parsing.

We take inspiration from recent research on universal dependency parsing (Tran and Bisazza, 2019; Kondratyuk and Straka, 2019). We employ an existing UD parsing framework — UDify, a multitask learning model (Kondratyuk and Straka, 2019) — and extend it to perform few-shot model adap-

★ Corresponding author: annalangedijk@gmail.com.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8503 - 8520

tation to previously unseen languages via meta-learning. We pre-train the dependency parser on a high-resource language prior to applying the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017) to a collection of few-shot tasks in a diverse set of languages. We evaluate our model on its ability to perform few-shot adaptation to unseen languages, from as few as 20 examples. Our results demonstrate that our methods outperform language transfer and multilingual joint learning baselines, as well as existing (zero-shot) UD parsing approaches, on a range of language families, with the most notable improvements among the low-resource languages. We also investigate the role of a pre-training language as a starting point for cross-lingual adaptation and the effect of typological properties on the learning process.

2 Related work

2.1 Meta-learning

In meta-learning, the datasets are separated into *episodes* that correspond to training tasks. Each episode contains a *support* and a *query* set, that include samples for adaptation and evaluation, respectively. Meta-learning serves as an umbrella term for algorithms from three categories: *Metric-based* methods classify new samples based on their similarity to the support set (e.g. Snell et al., 2017). *Model-based* methods explicitly store meta-knowledge within their architectures – e.g. through an external memory (Santoro et al., 2016). *Optimization-based* methods, on which we focus, estimate parameter initializations that can be fine-tuned with a few steps of gradient descent (e.g. Finn et al., 2017; Nichol and Schulman, 2018). Finn et al. (2017) proposed MAML to learn parameter initializations that generalize well to similar tasks. During the *meta-training* phase, MAML iteratively selects a batch of episodes, on which it fine-tunes the original parameters given the support set in an *inner learning loop*, and tests it on the query set. The gradients of the query set with respect to the original parameters are used to update those in the *outer learning loop*, such that these weights become a better parameter initialization over iterations. Afterwards, during *meta-testing*, one selects a support set for the test task, adapts the model using that set and evaluates it on new samples from the test task. MAML has provided performance benefits for cross-lingual transfer for tasks such as machine translation (Gu et al., 2018), named entity recognition (Wu et al., 2020), hypernymy detection (Yu et al., 2020) and mapping lemmas to inflected forms (Kann et al., 2020). The closest approach to ours is by Nooralahzadeh et al. (2020), who focus on natural language inference and question answering. Their method, X-MAML, involves pre-training a model on a high-resource language prior to applying MAML. This yielded performance benefits over standard supervised learning for cross-lingual transfer in a zero-shot and fine-tuning setup (albeit using 2500 training samples to fine-tune on test languages). The performance gains were the largest for languages sharing morphosyntactic features. Besides the focus on dependency parsing, our approach can be distinguished from Nooralahzadeh et al. (2020) in several ways. We focus on fast adaptation from a small number of examples (using only 20 to 80 sentences). Whilst they use one language for meta-training, we use seven languages, with the aim of explicitly learning to adapt to a variety of languages.

2.2 Universal dependency parsing

The Universal Dependencies project is an ongoing community effort to construct a cross-linguistically consistent morphosyntactic annotation scheme (Nivre et al., 2018). The project makes results comparable across languages and eases the evaluation of cross-lingual (structure) learning. The task of dependency parsing involves predicting a dependency tree for an input sentence, which is a directed graph of binary, asymmetrical arcs between words. These arcs are labeled and denote dependency relation types, which hold between a *head*-word and its *dependent*. A parser is tasked to assign rankings to the space of all possible dependency graphs and to select the optimal candidate.

Dependency parsing of under-resourced languages has since long been of substantial interest in NLP. Well-performing UD parsers, such as the winning model in the CoNLL 2018 Shared Task by Che et al. (2018), do not necessarily perform well on low-resource languages (Zeman et al., 2018). Cross-lingual UD parsing is typically accomplished by projecting annotations between languages with parallel corpora (Agić et al., 2014), through model transfer (e.g. Guo et al., 2015; Ammar et al., 2016; Ahmad et al., 2019), through hybrid methods combining annotation projections and model transfer (Tiedemann et al., 2014), or by aligning word embeddings across languages (Schuster et al., 2019).

State-of-the-art methods for cross-lingual dependency parsing exploit pre-trained mBERT with a dependency parsing classification layer that is fine-tuned on treebanks of high-resource languages, and transferred to new languages: Wu and Dredze (2019) only fine-tune on English, whereas Tran and Bisazza (2019) experiment with multiple sets of fine-tuning languages. Including diverse language families and scripts benefits transfer to low-resource languages, in particular. UDify, the model of Kondratyuk and Straka (2019), is jointly fine-tuned on data from 75 languages, with a multi-task learning objective that combines dependency parsing with predicting part-of-speech tags, morphological features, and lemmas. Üstün et al. (2020), instead, freeze the mBERT parameters and train adapter modules that are interleaved with mBERT's layers, and take a language embedding as input. This embedding is predicted from typological features. Model performance strongly relies on the availability of those features, since using proxy embeddings from different languages strongly degrades low-resource languages' performance.

3 Dataset

We use data from the Universal Dependencies v2.3 corpus (Nivre et al., 2018). We use treebanks from 26 languages that are selected for their typological diversity. We adopt the categorization of high-resource and low-resource languages from Tran and Bisazza (2019) and employ their set of training and test languages for comparability. The set covers languages from six language families (Indo-European, Korean, Afro-Asiatic, Uralic, Dravidian, Austro-Asiatic). Their training set (*expMix*) includes eight languages: English, Arabic, Czech, Hindi, Italian, Korean, Norwegian, and Russian. These languages fall into the language families of Indo-European, Korean and Afro-Asiatic and have diverse word orders (i.e. VSO, SVO and SOV). Joint learning on data from this diverse set yielded state-of-the-art zero-shot transfer performance on low-resource languages in the experiments of Tran and Bisazza (2019).

Per training language we use up to 20,000 example trees, predicting dependency arc labels from 132 classes total. We select Bulgarian (Indo-European) and Telugu (Dravidian) as validation languages to improve generalization to multiple language families. The 16 test languages cover three new language families that were unseen during training, i.e. Austro-Asiatic, Dravidian, and Uralic. Furthermore, three of our test languages (Buryat, Faroese, and Upper Sorbian) are not included in the pre-training of mBERT. We refer the reader to Appendix B for details about the treebank sizes and language families.

4 Method

4.1 The UDify model

The UDify model concurrently predicts part-of-speech tags, morphological features, lemmas and dependency trees (Kondratyuk and Straka, 2019). UDify exploits the pre-trained mBERT model (Devlin et al., 2019), that is a self-attention network with 12 transformer encoder layers.

The model takes single sentences as input. Each sentence is tokenized into subword units using mBERT's word piece tokenizer, after which contextual embedding lookup provides input for the self-attention layers. A weighted sum of the outputs of all layers is computed (Equation 1) and fed to a task-specific classifier.

$$e_j^t = \eta \sum_i \mathbf{B}_{ij} \cdot \mathrm{softmax}(\gamma)_i \qquad (1)$$

Here, e^t denotes the contextual output embeddings for task t. In our case, t indicates UD-parsing. In contrast to the multi-task objective of the original UDify model, our experiments only involve UD-parsing. The term $\mathbf{B}_{ij}$ represents the mBERT representation for layer $i = 1, ..., 12$ at token position j. The terms γ and η denote trainable scalars, where the former applies to mBERT and the latter scales the normalized averages. For words that were tokenized into multiple word pieces, only the first word piece was fed to the UD-parsing classifier.

The UD-parsing classifier is a graph-based biaffine attention classifier (Dozat and Manning, 2017) that projects the embeddings e_j^t through arc-head and arc-dep feedforward layers. The resulting outputs are combined using biaffine attention to produce a probability distribution of arc heads for each word. Finally, the dependency tree is decoded using the Chu-Liu/Edmonds algorithm (Chu, 1965; Edmonds, 1967). We refer the reader to the work of Kondratyuk and Straka (2019) for further details on the architecture and its training procedure.

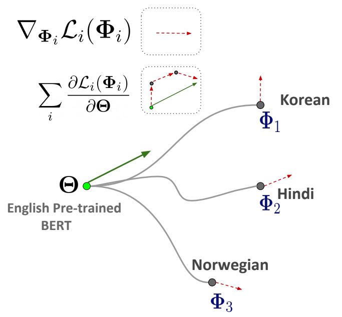

Figure 1: Visualization of MAML algorithm for three meta-training languages. The green arrows represents the meta-update from the outer learning loop. The red dotted arrows represent the gradient computed on the support set for each language in the inner learning loop.

4.2 Meta-learning procedure

We apply first-order[1] MAML to the UDify model. The model's self-attention layers are initialized with parameters from mBERT and the classifier's feedforward layers are randomly initialized. The model is pre-trained on a high-resource language using standard supervised learning and further meta-trained on a set of seven languages with MAML. It is then evaluated using meta-testing. We refer to MAML with pre-training as simply MAML. The meta-learning procedure is visualized in Figure 1 and can be described as follows:

Step 1 Pre-train on a high-resource language to yield the initial parameters Θ.

Step 2 Meta-train on all other training languages. For each language i, we partition the UD training data into two disjoint sets, D_i^{train} and D_i^{test}, and perform the following inner loop:

1. Temporarily update the model parameters Θ_i with stochastic gradient descent on support set S, sampled from D_i^{train}, with learning rate α for k gradient descent adaptation steps. When using a single gradient step, the update becomes:

$$\Phi_i \leftarrow \Theta - \alpha \nabla_\Theta \mathcal{L}(\Theta_i) \tag{2}$$

2. Compute the losses of the model parameters Φ_i using the query set Q, sampled from D_i^{test}, denoted by $\mathcal{L}_i(\Phi_i)$.

Step 3 Sum up the test losses and perform a meta-update in the outer learning loop on the model with parameters Θ using the learning rate β:

$$\Theta \leftarrow \Theta - \beta \sum_i \nabla_\Theta \mathcal{L}_i(\Phi_i) \tag{3}$$

In our experiments, the update is a first-order approximation, replacing $\nabla_\Theta \mathcal{L}_i(\Phi_i)$ by $\nabla_{\Phi_i} \mathcal{L}_i(\Phi_i)$.

Step 4 After meta-training, we apply meta-testing to unseen languages. For each language, we sample a support set S from the UD training data. We then fine-tune our model on S, and evaluate the model on the entire test set. Thereby, meta-testing mimics the adaptation from the inner loop. We repeat this process multiple times to get a reliable estimate of how well the model adapts to unseen languages.

5 Experimental setup

We extend the existing UDify code[2] to be used in a meta-learning setup. All of our code is publicly available.[3]

5.1 Training and evaluation

Pre-training In the main body of the paper, we consider the pre-training languages English and Hindi to measure the impact of pre-training prior to cross-lingual adaptation, and to draw more general conclusions about how well MAML generalizes with typologically different pre-training languages. English and Hindi differ in word order (SVO versus SOV), and Hindi treebanks have a larger percentage of non-projective dependency trees (Mannem et al., 2009), where dependency arcs are allowed to cross one another. Non-projective trees are more challenging to parse (Nivre, 2009). Pre-training on Hindi allows us to test the effects of projectivity on cross-lingual adaptation. To ensure that our findings are not specific to the pre-training languages of English and Hindi, Appendix D reproduces a subset of experiments for the pre-training languages Italian and Czech, reporting results for monolingual baselines, a non-episodic baseline, and MAML. Italian and Czech are high-resource languages as well, but are from two different subfamilies of the family of Indo-European languages and also differ in their percentage of non-projective dependency trees.

[1]For more details on first-order versus second-order, see Finn et al. (2017); Holla et al. (2020).

[2]github.com/Hyperparticle/udify

[3]github.com/annaproxy/udify-metalearning

Meta-training We apply meta-training using seven languages listed in Section 3, excluding the pre-training language from meta-training. We train for 500 episodes per language, using a cosine-based learning rate scheduler with 10% warm-up. We use the Adam optimizer (Kingma and Ba, 2015) in the outer loop and SGD in the inner loop (Finn et al., 2017). Support and query sets are of size 20. Due to the sequence labelling paradigm, the number of shots per class varies per batch. When $|S| = 20$, the average class will appear 16 times. To select hyperparameters, we independently vary the amount of updates k and the learning rates in the inner loop and outer loop for mBERT and the parser, while performing meta-validation with the languages Bulgarian and Telugu. To meta-validate, we follow the procedure described in Section 4.2 for both languages, mimicking the meta-testing setup with a support set size of 20. The hyperparameters are estimated independently for Hindi and English pre-training (see Appendix A).

Meta-testing At meta-testing time, we use SGD with the same learning rates and the same k used in the inner loop during meta-training. We vary the support set size $|S| \in \{20, 40, 80\}$.

5.2 Baselines

We define several baselines, that are evaluated using meta-testing, i.e. by fine-tuning the models on a support set of a test language prior to evaluation on that language. This allows us to directly compare their ability to *adapt quickly* to new languages with that of the meta-learner.

Monolingual baselines (EN, HIN) These baselines measure the impact of meta-training on data from seven additional languages. The model is initialized using mBERT and trained using data from English (EN) or Hindi (HIN), without meta-training.

Multilingual non-episodic baseline (NE) Instead of episodic training, this baseline treats support and query sets as regular mini-batches and updates the model parameters directly using a joint learning objective, similar to Kondratyuk and Straka (2019) and Tran and Bisazza (2019). The model is pre-trained on English or Hindi and thus indicates the advantages of MAML over standard supervised learning. The training learning rate and meta-testing learning rate are estimated separately, since there is no inner loop update in this setup.

MAML without pre-training We evaluate the effects of pre-training by running a MAML setup without any pre-training. Instead, the pre-training language is included during meta-training as one of now eight languages. MAML without pre-training is trained on 2000 episodes per language.

Meta-testing only The simplest baseline is a decoder randomly initialized on top of mBERT, without pre-training and meta-training. Dependency parsing is only introduced at meta-testing time.

5.3 Evaluation

Hyperparameter selection and evaluation is performed using Labeled Attachment Score (LAS) as computed by the CoNLL 2018 Shared Task evaluation script.[4] LAS evaluates the correctness of both the dependency class and dependency head. We use the standard splits of Universal Dependencies for training and evaluation when available. Otherwise, we remove the meta-testing support set from the test set prior to evaluation. We train each model with seven different seeds and compare MAML to a monolingual baseline and NE using paired t-tests, adjusting for multiple comparisons using Bonferroni correction.

6 Results and analysis

MAML with English pre-training We report the mean LAS for models pre-trained on English in Table 1. We compare these results to related approaches that use mBERT and have multiple training languages. With support set size 20, MAML already outperforms the zero-shot transfer setup of Tran and Bisazza (2019) for all test languages, except Persian and Urdu. MAML is competitive with UDify (Kondratyuk and Straka, 2019) and UDapter (Üstün et al., 2020) for low-resource languages, despite the stark difference in the number of training languages compared to UDify[5] (75), and without relying on fine-grained typological features of languages, as is the case for UDapter.

MAML consistently outperforms the EN and NE baselines. Large improvements over the EN baseline are seen on low-resource and non-Germanic languages. The difference between MAML and the baselines increases with $|S|$. The largest improvements over NE are on Tamil and Japanese,

[4] universaldependencies.org/conll18/evaluation.html

[5] UDify is trained on the low-resource languages, while we only test on them. For a fair comparison, we only list UDify results on languages with a small amount of sentences (<80) in the training set, to mimic a few-shot generalisation setup.

Language	T&B	K&S	Üst.	$\|S\| = 20$ EN	$\|S\| = 20$ NE	$\|S\| = 20$ MAML	$\|S\| = 40$ EN	$\|S\| = 40$ NE	$\|S\| = 40$ MAML	$\|S\| = 80$ EN	$\|S\| = 80$ NE	$\|S\| = 80$ MAML
Low-Resource Languages												
Armenian	58.95	–	–	49.8	63.34	**63.84**	50.59	63.54	**64.30**	51.99	63.79	**64.78**
Breton	52.62	39.84	58.5	60.34	61.44	**64.18**	61.32	61.67	**65.12**	62.76	62.20	**66.14**
Buryat†	23.11	26.28	**28.9**	23.66	25.56	**25.77**	23.82	25.67	**26.38**	24.17	25.88	**27.33**
Faroese†	61.98	59.26	69.2	68.50	67.83	**68.95**	69.56	68.12	**69.88**	70.59	68.62	**71.12**
Kazakh	44.56	**63.66**	**60.7**	47.25	55.02	**55.07**	47.80	55.08	**55.46**	49.08	55.23	**56.15**
U.Sorbian†	49.74	**62.82**	54.2	49.29	54.47	**56.40**	50.55	54.70	**57.55**	52.11	55.07	**58.81**
Mean	48.45	–	–	49.81	54.61	55.70	50.61	54.80	56.45	51.78	55.13	57.38
High-Resource Languages												
Finnish	62.29	–	–	56.61	**64.94**	64.89	56.99	65.07	**65.40**	57.73	65.18	**65.82**
French	59.54	–	–	65.21	66.55	**66.85**	65.33	66.59	**66.97**	65.63	66.65	**67.25**
German	70.93	–	–	72.47	76.15	**76.41**	72.6	76.17	**76.54**	72.93	76.21	**76.72**
Hungar.	61.11	–	–	56.50	**62.93**	62.71	56.23	**63.09**	62.81	56.73	**63.21**	62.52
Japanese	24.10	–	–	18.87	36.49	**39.06**	20.05	37.15	**42.17**	22.80	38.40	**46.81**
Persian	**56.92**	–	–	43.43	52.55	**52.81**	44.53	52.76	**53.63**	46.42	53.11	**54.74**
Swedish	78.70	–	–	80.26	80.73	**81.36**	80.41	80.81	**81.53**	80.57	80.79	**81.59**
Tamil	32.78	–	–	31.58	41.12	**44.34**	32.67	41.72	**46.73**	34.81	42.88	**50.73**
Urdu	**63.06**	–	–	25.71	**57.25**	55.16	26.89	**57.36**	56.16	29.30	**57.68**	57.60
Vietnam.	29.71	–	–	43.24	42.73	**43.34**	43.65	42.82	**43.74**	44.28	43.02	**44.34**
Mean	53.91	–	–	49.39	58.14	58.69	49.93	58.35	59.57	51.12	58.71	60.81
Mean	51.88	–	–	49.55	56.82	57.57	50.19	57.02	58.4	51.37	57.37	59.52

Table 1: Mean LAS aligned accuracy per support set size $|S|$ for unseen test languages. Best results per category are bolded. Significant results are underlined ($p < 0.005$). Previous work consists of Tran and Bisazza (2019), UDify (Kondratyuk and Straka, 2019) and UDapter (Üstün et al., 2020). †: Languages were absent from mBERT.

Language	$\|S\| = 20$ MAML	$\|S\| = 20$ MAML-	$\|S\| = 80$ MAML	$\|S\| = 80$ MAML-
Low-Resource Languages				
Armenian	**63.84**	59.70	**64.78**	60.03
Breton	**64.18**	59.33	**66.14**	60.84
Buryat†	25.77	**26.02**	**27.33**	27.05
Faroese†	**68.95**	65.30	**71.12**	66.79
Kazakh	**55.07**	53.92	**56.15**	54.99
U.Sorbian†	**56.40**	51.67	**58.78**	52.38
Mean	55.7	52.66	57.38	53.68
High-Resource Languages				
Mean	58.69	57.04	60.81	58.25

Table 2: Mean LAS per unseen language, for MAML without pre-training (denoted MAML-) versus MAML (EN). †: Languages were absent from mBERT.

however NE outperforms MAML on Hungarian and Urdu. MAML consistently outperforms NE on low-resource languages, with an average 1.1% improvement per low-resource language for $|S| = 20$, up to a 2.2% average improvement for $|S| = 80$.

MAML with Hindi pre-training The results for models pre-trained on Hindi can be seen in Table 3. Although there are large differences between the monolingual EN and HIN baselines, both MAML (HIN) and NE (HIN) achieve, on average, similar LAS scores to their English counterparts. MAML still outperforms NE for the majority of languages: the mean improvement on low-resource languages is 0.8% per language for $|S| = 20$, which increases to 1.6% per language for $|S| = 80$.

Other pre-training languages The full results for the two other pre-training languages, Italian and Czech, are listed in Appendix D. Here, too, MAML outperforms its NE counterpart. The NE baseline is stronger for more languages than in our main experiments. For $|S| = 20$, the mean improvements per unseen language are 0.91% and 0.47% when pre-training on Italian and Czech, respectively. For $|S| = 80$, the improvements are 2.18% and 1.75%.

MAML without (pre-)training We investigate the effectiveness of pre-training by omitting the pre-training phase. A comparison between MAML and MAML without pre-training is shown in Table 2. MAML without pre-training underperforms for most languages and its performance does not increase as much with a larger support set size. This suggests that pre-training provides a better starting point for meta-learning than plain mBERT.

When meta-testing only – i.e. omitting both pre-training and meta-training – the fine-tuned model reaches a mean LAS of 6.9% over all test languages for $|S| = 20$, increasing to 15% for $|S| = 80$, indicating that meta-testing alone is not sufficient

Language	$\lvert S\rvert = 20$			$\lvert S\rvert = 40$			$\lvert S\rvert = 80$		
	HIN	NE	MAML	HIN	NE	MAML	HIN	NE	MAML
Low-Resource Languages									
Armenian	48.41	63.30	**63.76**	48.87	63.41	**64.17**	49.70	63.59	**64.76**
Breton	34.06	**62.09**	61.56	36.09	62.40	**62.47**	38.95	63.05	**63.75**
Buryat†	24.24	25.05	**26.27**	24.71	25.18	**26.79**	25.54	25.40	**27.37**
Faroese†	50.72	65.31	**66.82**	52.30	65.57	**67.31**	54.64	66.17	**68.25**
Kazakh	49.80	53.77	**54.23**	49.90	53.94	**54.45**	50.49	54.08	**55.00**
U.Sorbian†	36.22	53.36	**54.97**	37.08	53.58	**55.64**	38.22	53.94	**56.56**
Mean	40.57	53.81	54.60	41.49	54.01	55.14	42.92	54.37	55.95
High-Resource Languages									
Finnish	50.49	64.05	**64.64**	50.93	64.20	**65.05**	51.79	64.40	**65.61**
French	31.16	64.44	**65.73**	31.59	64.44	**65.68**	33.39	64.42	**65.69**
German	44.83	74.40	**75.15**	45.46	74.41	**75.23**	46.65	74.46	**75.31**
Hungarian	46.72	60.98	**62.51**	46.97	61.33	**62.89**	47.91	61.68	**62.91**
Japanese	40.25	39.97	**41.96**	43.03	40.56	**43.61**	**46.87**	41.58	45.90
Persian	28.60	**53.73**	53.63	29.51	53.85	**54.00**	31.11	54.06	**54.53**
Swedish	46.96	79.24	**79.89**	47.73	79.32	**80.14**	49.15	79.31	**80.21**
Tamil	**46.51**	39.44	39.57	**47.35**	39.84	40.84	**48.55**	40.73	42.81
Urdu	**67.72**	50.64	49.16	**67.96**	50.93	50.16	**68.17**	51.50	51.57
Vietnamese	26.96	**42.13**	42.12	27.92	42.23	**42.37**	29.61	42.46	**42.87**
Mean	43.02	56.9	57.44	43.85	57.11	58.0	45.32	57.46	58.74
Mean	42.1	55.74	56.37	42.96	55.95	56.92	44.42	56.3	57.69

Table 3: Mean LAS aligned accuracy per unseen language, for models pre-trained on Hindi. Best results per category are listed in bold, significant results are underlined ($p < 0.005$). †: Languages were absent from mBERT.

to learn the task.[6]

Further Analysis Performance increases over the monolingual baselines vary strongly per language – e.g. consider the difference between Japanese and French in Table 1. The performance increase is largest for languages that differ from the pre-training language with respect to their syntactic properties. We conduct two types of analysis, based on typological features and projectivity, to quantify this effect and correlate these properties to the performance increase over monolingual baselines.[7]

Firstly, we use 103 binary syntactic features from URIEL (Littell et al., 2017) to compute the syntactic cosine similarities (denoted σ) between languages. With this metric, a language such as Italian is syntactically closer to English ($\sigma = 0.86$) than Urdu ($\sigma = 0.62$), even though they are both Indo-European. For each unseen language, we collect the cosine similarities to each (pre-)training language. Then, we collect the difference in performance between the monolingual baselines and the NE or MAML setups for $|S| = 20$. For each training language, we compute the correlations between performance increases for the test languages and their similarity to this training language, visualised in Figure 2. When pre-training on Hindi, there is a significant positive correlation with syntactic similarity to English and related languages. When pre-training on English, a positive correlation is seen with similarity to Hindi and Korean. Positive correlations imply that on unseen languages, improvement increases when similarity to the training language increases. Negative correlations mean there is less improvement when similarity to the training languages increases, suggesting that those languages do not contribute as much to adaptation. On average, the selection of meta-training languages contributes significantly to the increase in performance for the Hindi pre-training models. This effect is stronger for MAML (HIN) ($p = 0.006$) than NE (HIN) ($p = 0.026$), which may indicate that the meta-training procedure is better at incorporating knowledge from those unrelated languages.

Secondly, we analyze which syntactic features impact performance most. We correlate individual URIEL features with MAML's performance increases over monolingual baselines (see Figure 3). Features related to word order and negation show a significant correlation. Considering the presence of these features in both pre-training languages of MAML, a pattern emerges: when a feature is absent in the pre-training language, there is a positive correlation with increase in performance. Similarly, when a feature is present in the pre-training lan-

[6]Full results can be found in Appendix C.

[7]No clear correlation was found by Tran and Bisazza (2019). By using *increase in performance* and not "plain" performance, we may see a stronger effect.

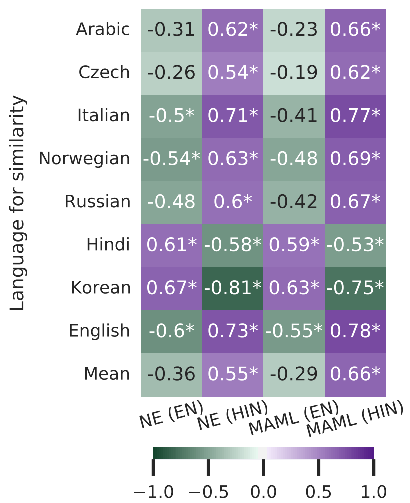

Figure 2: Spearman's ρ between the performance increase over the monolingual baseline and the cosine similarity to the syntax of training languages (y-axis) for models using pre-training (x-axis). (*: $p < 0.05$)

guage, there is a negative correlation, and thus a smaller increase in performance after meta-training. This indicates that MAML is successfully adapting to these specific features during meta-training.

We analyzed MAML's performance improvements over NE on each of the 132 dependency relations, and found that they are consistent across relations. [8] Lastly, we detect non-projective dependency trees in all datasets. The Hindi treebank used has 14% of non-projective trees, whereas English only has 5%.[9] We correlate the increase in performance with the percentage of non-projective trees in a language's treebank. The correlation is significant for NE (EN) (ρ = 0.46, p = 0.01) and MAML (EN) (ρ = 0.42, p = 0.03). Figure 4 visualizes the correlation for MAML (EN). We do not find significant correlations for models pre-trained on Hindi. This suggests that a model trained on a mostly projective language can benefit more from further training on non-projective languages than the other way around. The same trend is observed when comparing models pre-trained on Italian and Czech, that also differ in the percentage of non-projective trees (Appendix D).

7 Discussion

Our experiments confirm that meta-learning, specifically MAML, is able to adapt to unseen languages on the task of cross-lingual dependency parsing

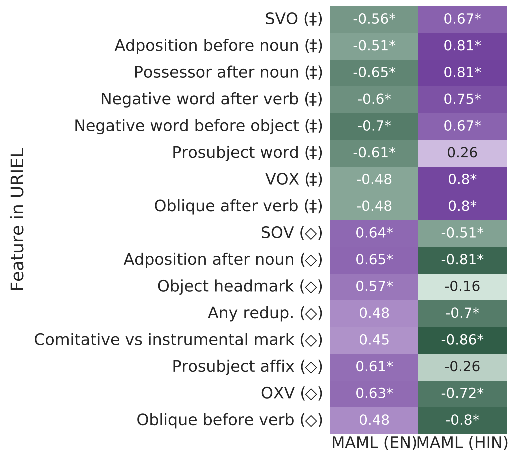

Figure 3: Spearman's ρ between the performance increase over monolingual baselines and URIEL features (y-axis), for MAML (x-axis). We indicate features present in English (‡) and in Hindi (◇). (*: $p < 0.05$)

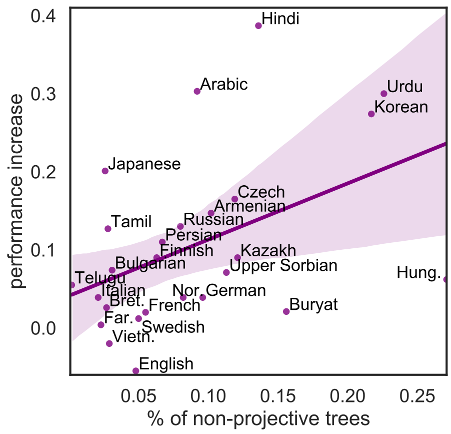

Figure 4: Spearman's ρ between the % of non-projective dependency trees and MAML's improvement over the English baseline (ρ = 0.42, p = 0.03).

more effectively than a non-episodic model. The difference between both methods is most apparent for languages that differ strongly from those in the training set (e.g. Japanese in Table 1) where effective few-shot adaptation is crucial. This shows that MAML is successful at *learning to learn* from a few examples, and can efficiently incorporate new information. Furthermore, we see a clear increase in performance for MAML when increasing the test support set size, while NE only slightly improves. This suggests that MAML may be a promising method for cross-lingual adaptation more generally, also outside of the few-shot learning scenario.

Our ablation experiments on pre-training show that it is beneficial for MAML to start from a strong set of parameters, pre-trained on a high resource

[8] The same holds for the 37 coarse-grained UD relations.
[9] Full results can be found in Appendix B.

language. Thereby, the pre-training is not dependent on a specific language. MAML performs well with a variety of pre-training languages, although improvements for unseen languages vary. When a model is pre-trained on English, there is a large positive correlation for improvements in languages that are syntactically dissimilar to English, such as Japanese and Tamil. During meta-training, dissimilar training languages such as Hindi most contribute to the model's ability to generalize. Syntactic features, especially those related to word order, which have already been learned during pre-training, require less adaptation. The same is true, vice versa, for Hindi pre-training.

This effect is also observed, though only in one direction, when correlating performance increase with non-projectivity. It is beneficial to meta-train on a set of languages that vary in projectivity after pre-training on one which is mostly projective. However, not all variance is explained by the difference in typological features. The fact that MAML outperforms MAML without pre-training suggests that pre-training also contributes *language-agnostic* syntactic features, which is indeed the overall goal of multi-lingual UD models.

8 Conclusion

In this paper, we present a meta-learning approach for the task of cross-lingual dependency parsing. Our experiments show that meta-learning can improve few-shot universal dependency parsing performance on unseen, unrelated test languages, including low-resource languages and those not covered by mBERT. In addition, we see that it is beneficial to pre-train before meta-training, as in the X-MAML approach (Nooralahzadeh et al., 2020). In particular, the pre-training language can affect how much adaptation is necessary on languages that are typologically different from it. Therefore, an important direction for future research is to investigate a wider range of pre-training/meta-training language combinations, based on specific hypotheses about language relationships and relevant syntactic features. Task performance may be further improved by including a larger set of syntax-related tasks, such as POS-tagging, to sample from during meta-training (Kondratyuk and Straka, 2019).

References

Željko Agić, Jörg Tiedemann, Kaja Dobrovoljc, Simon Krek, Danijela Merkler, and Sara Može. 2014. Cross-lingual dependency parsing of related languages with rich morphosyntactic tagsets. In *EMNLP 2014 Workshop on Language Technology for Closely Related Languages and Language Variants*.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4517–4533, Online. Association for Computational Linguistics.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-Learning in Neural Networks: A Survey.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 732–738, New Orleans, Louisiana. Association for Computational Linguistics.

Katharina Kann, Samuel R Bowman, and Kyunghyun Cho. 2020. Learning to Learn Morphological Inflection for Resource-Poor Languages. In *AAAI*, pages 8058–8065.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. 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 2779–2795.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. 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.

Shangwen Lv, Yuechen Wang, Daya Guo, Duyu Tang, Nan Duan, Fuqing Zhu, Ming Gong, Linjun Shou, Ryan Ma, Daxin Jiang, et al. 2020. Pre-training text representations as meta learning. *arXiv preprint arXiv:2004.05568*.

Prashanth Mannem, Himani Chaudhry, and Akshar Bharati. 2009. Insights into non-projectivity in Hindi. *ACL-IJCNLP 2009 - Joint Conf. of the 47th Annual Meeting of the Association for Computational Linguistics and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Proceedings of the Conf.*, (August):10–17.

Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 351–359.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2018. Universal Dependencies 2.3. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

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.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140.

Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288.

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.

Qianhui Wu, Zijia Lin, Guoxin Wang, Hui Chen, Börje F Karlsson, Biqing Huang, and Chin-Yew Lin. 2020. Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. In *AAAI*, pages 9274–9281.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Changlong Yu, Jialong Han, Haisong Zhang, and Wilfred Ng. 2020. Hypernymy detection for low-resource languages via meta learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3656.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.

A Training details and hyperparameters

Parameter	Value
Dependency tag dimension	256
Dependency arc dimension	768
Dropout	0.5
BERT Dropout	0.2
Mask probability	0.2
Layer dropout	0.1

Table 4: Hyperparameters for the UDify model architecture.

All models use the same architecture: an overview of all model parameters can be seen in Table 4. The model contains 196M parameters, of which 178M are mBERT.

At pre-training time, we use the default parameters of UDify (Kondratyuk and Straka, 2019). We pre-train for 60 epochs. The Adam optimizer is used with a 1e-3 learning rate for the decoder and a 5e-5 learning rate for BERT layers. Weight decay of 0.01 is applied. We employ a gradual unfreezing scheme, freezing the BERT layer weights for the first epoch.

At meta-training time, we vary the learning rates as shown in Table 5. We also vary the amount of updates k at training/testing time: $k \in \{8, 20\}$. We applied weight decay at meta-training time in initial experiments, but this yielded no improvements. No gradual unfreezing is applied at meta-training time. We use Adam for the outer loop updates and SGD for the inner loop updates and at testing time. We sample 500 episodes per language, using query and support set size of 20. The best hyperparameters are chosen with respect to their final performance on the meta-validation set consisting of Bulgarian and Telugu. We run two seeds for hyperparameter selection, and seven seeds for all the final models. Labeled Attachment Score (LAS) is used for hyperparameter selection and final evaluation.

We train all models on an NVIDIA TITAN RTX. Pre-training takes around 3 hours, meta-training takes around 1 hour for 100 episodes per language when the amount of updates k is set to 20. For MAML, this amounts to approximately 5 hours, and for the ablated MAML-, it amounts to approximately 20 hours, which can be seen as another benefit of pre-training. Finally, training in a non-episodic fashion (NE) also takes up less time, namely 2 to 3 hours.

All final learning rates can be seen in Table 6. For all models except the random decoder baseline, $k = 20$ was selected. The best random decoder used $k = 80$ after a separate hyperparameter search of high learning rates and ks (compensating for the lack of prior DP training).

LR	**mBERT**	**Decoder**
Inner/Test α	{1e-4, <u>5e-5</u>, <u>1e-5</u>}	{<u>1e-3</u>, 5e-4, <u>1e-4</u>, 5e-5}
Outer β	{<u>5e-5</u>, <u>1e-5</u>, 7e-6}	{<u>1e-3</u>, <u>7e-4</u>, <u>5e-4</u>, <u>1e-4</u>, 5e-5}

Table 5: Learning rates independently varied for MAML and NE. For the ablated MAML-, only <u>underlined</u> learning rates were tried due to the long training times.

	Inner/Test LR		**Outer LR**	
	Decoder	BERT	Decoder	BERT
Meta-test only	5e-3	1e-3	n/a	n/a
EN/HIN	1e-4	1e-4	n/a	n/a
NE (en/hin)	5e-4	1e-5	1e-4	7e-6
MAML (en)	1e-3	1e-4	5e-4	1e-5
MAML (hin)	5e-4	5e-5	5e-4	5e-5
MAML-	1e-3	1e-5	5e-4	1e-5

Table 6: Final hyperparameters, as selected by few-shot performance on the validation set. Inner loop/Test learning rates are used with SGD, outer loop LRs are used with the Adam optimizer.

B Information about datasets used

All information about the datasets used can be found in Table 7, along with corresponding statistics about non-projective trees. The cosine syntactical similarities are visualized in Figure 5.

C Full results

We show the full results for each model in Table 8, Table 9 and Table 10.

Language	Family	Subcategory	UD Dataset	Train	Val.	Test	Non-proj. %
Low-Resource Test Languages							
Armenian	IE	Armenian	ArmTDP	560	0	470	10.2
Breton	IE	Celtic	KEB	0	0	888	2.7
Buryat	Mongolic	Mongolic	BDT	19	0	908	15.6
Faroese	IE	Germanic	OFT	0	0	1208	2.7
Kazakh	Turkic	Northwestern	KTB	31	0	1047	12.1
Upper Sorbian	IE	Slavic	UFAL	23	0	623	11.3
High-Resource Test Languages							
Finnish	Uralic	Finnic	TDT	12217	1364	1555	6.3
French	IE	Romance	Spoken	1153	907	726	5.5
German	IE	Germanic	GSD	13814	799	977	9.2
Hungarian	Uralic	Ugric	Szeged	910	441	449	27.1
Japanese	Japanese	Japanese	GSD	7133	511	551	2.6
Persian	IE	Iranian	Seraji	4798	599	600	6.7
Swedish	IE	Germanic	PUD	0	0	1000	3.8
Tamil	Dravidian	Southern	TTB	400	80	120	2.8
Urdu	IE	Indic	UDTB	4043	552	535	22.6
Vietnamese	Austro-As.	Viet-Muong	VTB	1400	800	800	2.9
Validation Languages							
Bulgarian	IE	Slavic	BTB	8907	1115	1116	3.1
Telugu	Dravidian	South Central	MTG	1051	131	146	0.2
Train Languages							
Arabic	Afro-As.	Semitic	PADT	6075	909	680	9.2
Czech	IE	Slavic	PDT	68495	9270	10148	11.9
English	IE	Germanic	EWT	12543	2002	2077	4.8
Hindi	IE	Indic	HDTB	13304	1659	1684	13.6
Italian	IE	Romance	ISDT	13121	564	482	2.1
Korean	Korean	Korean	Kaist	23010	2066	2287	21.7
Norwegian	IE	Germanic	Nynorsk	14174	1890	1511	8.2
Russian	IE	Slavic	SynTagRus	48814	6584	6491	8.0

Table 7: All datasets used during testing (first 16 rows) training and evaluation (final 10 rows), along with the amount of sentences in the dataset and the percentage of non-projective trees throughout that dataset.

Language	M.T. only	EN	HIN	NE (EN)	NE (HIN)	MAML	MAML (HIN)	MAML-
Unseen Languages								
Armenian	4.97±0.007	49.8±0.005	48.41±0.002	63.34±0.002	63.3±0.005	**63.84**±0.002	63.76±0.003	59.7±0.004
Breton	10.77±0.019	60.34±0.003	34.06±0.005	61.44±0.005	62.09±0.004	**64.18**±0.003	61.56±0.002	59.33±0.005
Buryat	9.63±0.018	23.66±0.002	24.24±0.002	25.56±0.003	25.05±0.003	25.77±0.002	**26.27**±0.002	26.02±0.004
Faroese	13.86±0.024	68.5±0.004	50.72±0.004	67.83±0.006	65.31±0.006	**68.95**±0.003	66.82±0.002	65.3±0.005
Kazakh	13.97±0.012	47.25±0.004	49.8±0.002	55.02±0.002	53.77±0.003	**55.07**±0.002	54.23±0.003	53.92±0.005
U.Sorbian	3.44±0.005	49.29±0.004	36.22±0.003	54.47±0.003	53.36±0.003	**56.4**±0.004	54.97±0.005	51.67±0.004
Finnish	6.95±0.014	56.61±0.002	50.49±0.003	**64.94**±0.003	64.05±0.004	64.89±0.003	64.64±0.002	61.97±0.005
French	6.81±0.011	65.21±0.001	31.16±0.003	66.55±0.001	64.44±0.002	**66.85**±0.001	65.73±0.001	63.42±0.003
German	7.52±0.012	72.47±0.001	44.83±0.004	76.15±0.002	74.4±0.002	**76.41**±0.002	75.15±0.001	74.38±0.003
Hungarian	5.58±0.004	56.5±0.003	46.72±0.004	**62.93**±0.003	60.98±0.002	62.71±0.003	62.51±0.004	58.47±0.002
Japanese	4.02±0.008	18.87±0.002	40.25±0.005	36.49±0.008	39.97±0.003	39.06±0.003	**41.96**±0.005	39.72±0.007
Persian	1.91±0.004	43.42±0.005	28.66±0.004	52.62±0.006	**53.78**±0.004	52.82±0.005	53.59±0.003	50.31±0.004
Swedish	5.15±0.008	80.26±0.001	46.96±0.004	80.73±0.001	79.24±0.002	**81.36**±0.001	79.89±0.001	77.57±0.002
Tamil	5.18±0.013	31.58±0.005	46.51±0.003	41.12±0.009	39.44±0.006	44.34±0.005	39.57±0.008	**46.55**±0.01
Urdu	2.86±0.01	25.71±0.004	**67.72**±0.001	57.25±0.004	50.64±0.004	55.16±0.004	49.16±0.002	55.4±0.003
Vietnamese	7.14±0.008	43.24±0.002	26.96±0.002	42.73±0.001	42.13±0.002	**43.34**±0.001	42.12±0.001	42.62±0.003
Validation & Training Languages								
Bulgarian	8.65±0.01	71.19±0.002	46.76±0.003	78.42±0.003	77.62±0.001	**78.64**±0.002	78.4±0.001	75.3±0.003
Telugu	42.36±0.078	64.39±0.018	66.78±0.014	68.5±0.006	64.8±0.008	**69.91**±0.01	65.8±0.008	67.58±0.008
Arabic	3.25±0.007	38.53±0.006	20.74±0.004	71.51±0.002	69.76±0.002	68.86±0.002	**73.09**±0.002	66.4±0.002
Czech	6.37±0.006	67.3±0.002	43.24±0.002	83.15±0.001	81.65±0.001	82.0±0.001	**83.21**±0.001	80.06±0.001
English	8.43±0.008	**89.29**±0.001	44.48±0.003	82.15±0.004	79.48±0.001	83.74±0.001	81.89±0.001	78.04±0.001
Hindi	3.38±0.007	35.42±0.002	**90.99**±0.0	76.56±0.002	74.03±0.004	74.15±0.003	71.33±0.003	74.48±0.004
Italian	7.15±0.008	82.5±0.001	36.86±0.006	**87.34**±0.002	85.28±0.001	86.5±0.001	87.18±0.002	83.09±0.002
Korean	7.82±0.011	36.44±0.004	40.3±0.002	66.35±0.003	68.04±0.003	63.93±0.003	**74.08**±0.001	63.62±0.005
Norwegian	5.68±0.013	74.7±0.001	43.7±0.003	80.09±0.001	77.65±0.001	78.67±0.001	**81.33**±0.002	75.61±0.001
Russian	6.76±0.013	68.94±0.003	47.29±0.005	80.96±0.001	79.41±0.001	79.93±0.001	**81.68**±0.001	76.48±0.002

Table 8: Full meta-testing results for all models and baselines, including validation and training languages, for $|S| = 20$. The meta-testing only baseline is denoted as “M.T. only".

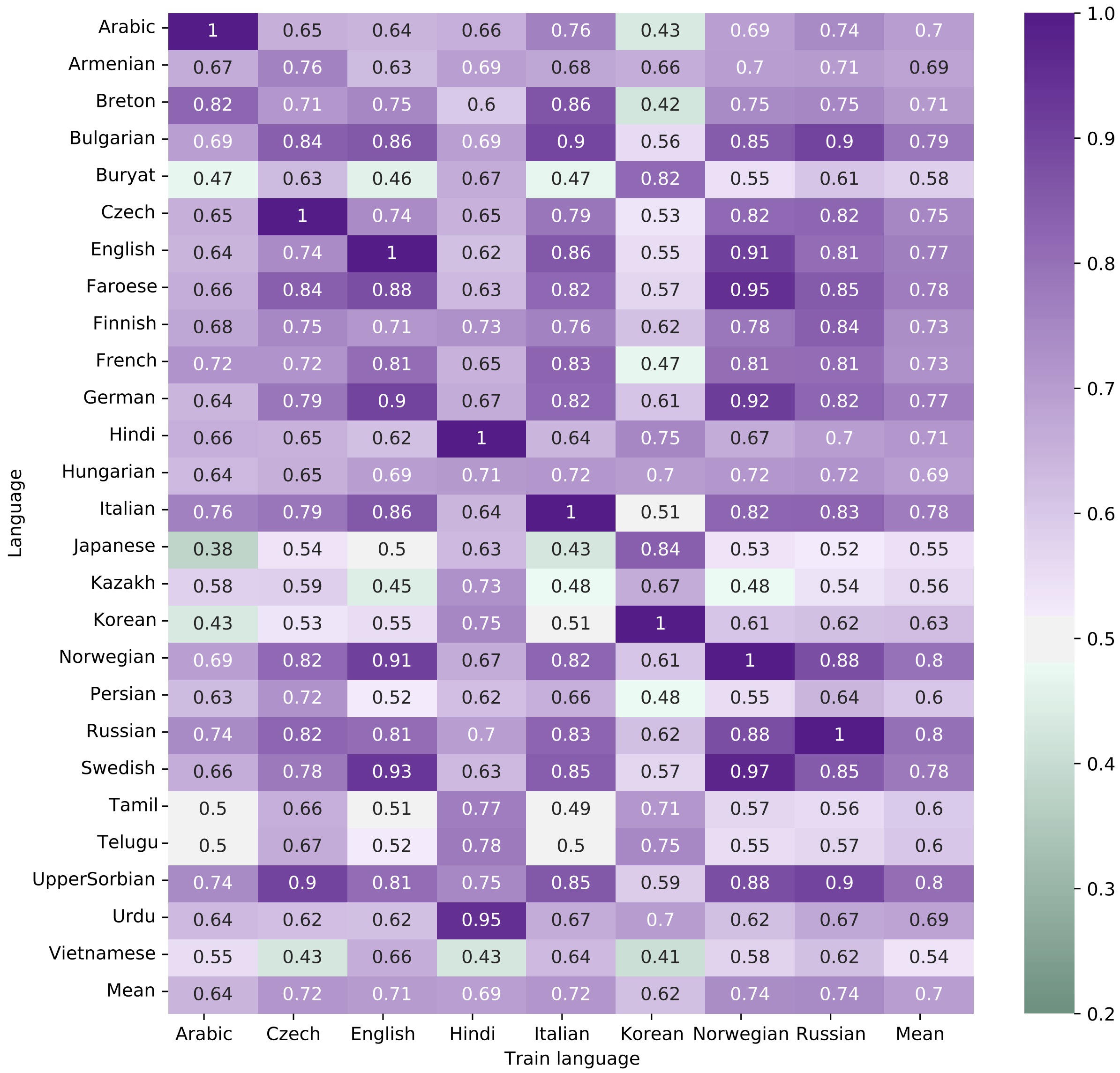

Figure 5: Syntactical cosine similarities from each training language to all other languages, calculated using URIEL's 103 binary syntactic features (Littell et al., 2017). Average cosine similarities are shown in the rightmost column and the bottom row. For instance, Japanese and Kazakh have a relatively low average cosine similarity to the training languages.

Language	M.T. only	EN	HIN	NE (EN)	NE (HIN)	MAML	MAML (HIN)	MAML-
Unseen Languages								
Armenian	5.82±0.007	50.59±0.005	48.87±0.002	63.54±0.002	63.41±0.004	**64.3**±0.002	64.17±0.003	59.85±0.004
Breton	14.52±0.02	61.32±0.004	36.09±0.005	61.67±0.005	62.4±0.005	**65.12**±0.003	62.47±0.004	59.96±0.004
Buryat	13.36±0.017	23.82±0.002	24.71±0.003	25.67±0.003	25.18±0.003	26.38±0.003	**26.79**±0.003	26.49±0.004
Faroese	20.4±0.019	69.56±0.004	52.3±0.005	68.12±0.006	65.57±0.006	**69.88**±0.004	67.31±0.003	65.95±0.004
Kazakh	17.11±0.014	47.8±0.004	49.9±0.003	55.08±0.002	53.94±0.003	**55.46**±0.003	54.45±0.004	54.29±0.005
U.Sorbian	4.49±0.008	50.55±0.005	37.08±0.003	54.7±0.004	53.58±0.003	**57.55**±0.004	55.64±0.005	52.09±0.005
Finnish	9.42±0.011	56.99±0.003	50.93±0.003	**65.07**±0.003	64.2±0.004	65.4±0.003	65.05±0.003	62.26±0.004
French	8.41±0.019	65.33±0.002	31.59±0.005	66.59±0.001	64.44±0.002	**66.97**±0.001	65.68±0.002	63.78±0.003
German	10.4±0.016	72.6±0.001	45.46±0.004	76.17±0.002	74.41±0.002	**76.54**±0.002	75.23±0.002	74.53±0.003
Hungarian	6.8±0.007	56.23±0.003	46.97±0.004	**63.09**±0.003	61.33±0.002	62.81±0.002	62.89±0.003	58.09±0.003
Japanese	5.85±0.014	20.05±0.003	43.03±0.006	37.15±0.008	40.56±0.002	42.17±0.004	**43.61**±0.004	41.51±0.006
Persian	3.32±0.01	44.54±0.004	29.55±0.004	52.72±0.006	**53.85**±0.004	53.65±0.005	54.02±0.003	50.83±0.005
Swedish	8.27±0.008	80.41±0.001	47.73±0.003	80.81±0.001	79.32±0.002	**81.53**±0.001	80.14±0.002	77.94±0.002
Tamil	9.37±0.018	32.67±0.004	47.35±0.004	41.72±0.009	39.84±0.004	46.73±0.005	40.84±0.006	**48.54**±0.008
Urdu	5.88±0.01	26.89±0.005	**67.96**±0.002	57.36±0.004	50.93±0.004	56.16±0.004	50.16±0.004	55.84±0.003
Vietnamese	9.65±0.014	43.65±0.002	27.92±0.002	42.82±0.002	42.23±0.003	**43.74**±0.001	42.37±0.001	43.23±0.004
Validation & Training Languages								
Bulgarian	10.87±0.021	71.21±0.002	47.29±0.004	78.42±0.003	77.62±0.001	**78.65**±0.002	78.39±0.001	75.44±0.003
Telugu	49.11±0.068	66.64±0.014	67.7±0.013	68.69±0.006	64.75±0.01	**70.75**±0.012	66.1±0.009	67.97±0.007
Arabic	4.83±0.006	41.6±0.012	22.65±0.005	71.53±0.002	69.78±0.002	68.95±0.002	**73.01**±0.003	66.47±0.002
Czech	7.95±0.008	67.74±0.003	43.92±0.002	83.15±0.001	81.64±0.001	82.03±0.001	**83.19**±0.001	80.12±0.001
English	11.01±0.012	**89.3**±0.001	45.32±0.004	82.21±0.004	79.49±0.001	83.96±0.002	82.05±0.002	78.07±0.001
Hindi	7.64±0.017	36.64±0.002	**90.99**±0.0	76.58±0.002	74.24±0.004	74.28±0.002	72.16±0.004	74.5±0.004
Italian	9.3±0.014	82.68±0.001	38.61±0.007	**87.35**±0.002	85.28±0.001	86.51±0.001	87.37±0.003	83.11±0.002
Korean	10.26±0.014	36.77±0.004	40.74±0.003	66.4±0.003	68.07±0.003	64.23±0.003	**74.01**±0.002	63.91±0.004
Norwegian	9.24±0.015	74.7±0.002	44.1±0.006	80.06±0.002	77.53±0.004	78.69±0.001	**81.2**±0.004	75.64±0.001
Russian	9.09±0.012	69.3±0.003	48.03±0.005	80.98±0.001	79.43±0.001	80.0±0.001	**81.66**±0.001	76.57±0.002

Table 9: Full meta-testing results for all models and baselines, including validation and training languages, for $|S| = 40$. The meta-testing only baseline is denoted as “M.T. only".

Language	M.T. only	EN	HIN	NE (EN)	NE (HIN)	MAML	MAML (HIN)	MAML-
Unseen Languages								
Armenian	8.19±0.006	51.99±0.005	49.7±0.002	63.79±0.002	63.59±0.004	**64.78**±0.003	64.76±0.003	60.03±0.003
Breton	22.54±0.018	62.76±0.004	38.95±0.004	62.2±0.006	63.05±0.004	**66.14**±0.003	63.75±0.004	60.84±0.004
Buryat	16.87±0.007	24.17±0.003	25.54±0.003	25.88±0.003	25.4±0.003	27.33±0.003	**27.37**±0.004	27.05±0.004
Faroese	27.76±0.019	70.59±0.004	54.64±0.005	68.62±0.006	66.17±0.005	**71.12**±0.004	68.25±0.003	66.79±0.004
Kazakh	21.89±0.009	49.08±0.004	50.49±0.003	55.23±0.003	54.08±0.003	**56.15**±0.003	55.0±0.004	54.99±0.005
U.Sorbian	7.49±0.01	52.11±0.005	38.22±0.004	55.08±0.004	53.94±0.004	**58.78**±0.005	56.56±0.006	52.38±0.005
Finnish	11.91±0.012	57.73±0.004	51.79±0.002	**65.18**±0.003	64.4±0.004	65.82±0.005	65.61±0.004	62.47±0.004
French	12.42±0.026	65.63±0.002	33.39±0.006	66.65±0.001	64.42±0.002	**67.25**±0.002	65.69±0.003	64.15±0.003
German	16.57±0.017	72.93±0.002	46.65±0.003	76.21±0.002	74.46±0.002	**76.72**±0.002	75.31±0.002	74.72±0.003
Hungarian	13.0±0.013	56.73±0.003	47.91±0.003	**63.21**±0.003	61.68±0.002	62.52±0.002	62.91±0.002	57.48±0.004
Japanese	14.38±0.015	22.8±0.004	46.87±0.004	38.4±0.007	41.58±0.003	46.81±0.003	**45.9**±0.004	43.87±0.005
Persian	6.16±0.019	46.4±0.006	31.11±0.01	53.08±0.006	**54.01**±0.004	54.73±0.006	54.54±0.005	51.07±0.004
Swedish	12.99±0.011	80.57±0.001	49.15±0.002	80.79±0.002	79.31±0.002	**81.59**±0.001	80.21±0.002	78.1±0.002
Tamil	18.46±0.011	34.81±0.007	48.55±0.002	42.88±0.008	40.73±0.004	50.68±0.003	42.81±0.006	**50.54**±0.008
Urdu	13.06±0.01	29.3±0.004	**68.17**±0.004	57.63±0.004	51.5±0.004	57.6±0.004	51.57±0.004	56.28±0.004
Vietnamese	15.36±0.015	44.28±0.002	29.61±0.002	42.99±0.002	42.46±0.003	**44.33**±0.002	42.88±0.002	43.78±0.004
Validation & Training Languages								
Bulgarian	16.26±0.025	71.42±0.003	48.07±0.006	78.43±0.003	77.67±0.002	**78.67**±0.003	78.45±0.002	75.68±0.003
Telugu	54.48±0.016	69.08±0.011	68.79±0.01	68.97±0.006	65.05±0.009	**71.52**±0.012	66.86±0.008	68.41±0.008
Arabic	9.87±0.015	46.24±0.015	25.5±0.006	71.54±0.002	69.79±0.002	69.07±0.002	**73.04**±0.002	66.51±0.002
Czech	10.74±0.012	68.4±0.003	45.28±0.002	83.16±0.001	81.65±0.001	82.04±0.001	**83.2**±0.001	80.15±0.001
English	16.86±0.016	**89.3**±0.001	46.87±0.002	82.32±0.003	79.51±0.001	84.28±0.002	82.08±0.002	78.07±0.002
Hindi	16.7±0.018	39.25±0.003	**90.96**±0.0	76.61±0.002	74.65±0.004	74.46±0.003	73.3±0.003	74.63±0.004
Italian	16.86±0.027	82.96±0.001	41.8±0.009	**87.35**±0.002	85.29±0.001	86.57±0.002	87.39±0.003	83.17±0.002
Korean	15.16±0.017	37.77±0.005	41.53±0.003	66.46±0.003	68.16±0.003	64.36±0.004	**74.05**±0.002	64.21±0.005
Norwegian	13.08±0.012	74.93±0.002	45.3±0.004	80.08±0.002	77.56±0.004	78.76±0.001	**81.22**±0.004	75.69±0.001
Russian	13.37±0.012	69.79±0.003	49.02±0.004	81.01±0.001	79.45±0.001	80.04±0.001	**81.67**±0.001	76.56±0.002

Table 10: Full meta-testing results for all models and baselines, including validation and training languages, for $|S| = 80$. The meta-testing only baseline is denoted as “M.T. only".

D Results for Italian/Czech pre-training

We repeat the setup that is discussed in the main paper on another pair of pre-training languages: Italian and Czech. These two languages are, as are English and Hindi, Indo-European, but vary in amount of non-projective sentences within their UD treebanks: 2.1% sentences are non-projective for the Italian UD dataset used, and 11.9% for the Czech dataset (see also Table 7). This allows us to further corroborate our findings on non-projectivity.

We randomly take 13 thousand sentences from the Czech training set to match the size of the other three pre-training sets used and verify that the percentage of non-projective sentences is of the same magnitude on this new training set. We run a separate, smaller hyperparameter search for these experiments. All hyperparameters for the monolingual (CZ, IT), non-episodic (NE), and meta-learning (MAML) models are selected using meta-validation. These hyperparameters can be seen in Table 11.

	Inner/Test LR		**Outer LR**	
	Decoder	BERT	Decoder	BERT
IT/CZ	1e-4	1e-4	n/a	n/a
NE (it/cz)	5e-4	1e-4	1e-4	7e-6
MAML (it/cz)	1e-3	5e-4	5e-4	1e-5

Table 11: Final hyperparameters in the Italian/Czech setup, as selected by few-shot performance on the meta-validation set. Inner loop/Test learning rates are used with SGD, outer loop LRs are used with the Adam optimizer.

D.1 Performance

The full results can be seen in Table 12 and Table 13. The results are similar to those in Table 1 for both the low-resource and the high-resource category. MAML slightly outperforms the corresponding non-episodic baseline NE, especially on unrelated languages from Italian and Czech, such as Japanese.

D.2 Projectivity

For MAML with Italian pre-training, Spearman's $\rho = 0.43$ ($p = 0.028$). For MAML with Czech pre-training, the effect is not significant $\rho = 0.3$ ($p = 0.1349$).

These correlations were, as in the original experiments, calculated using the training language set as well as the testing language set. Excluding training languages in this calculation, the correlation is weaker for Italian pre-training $\rho = 0.39, p = 0.048$ and non-existent for Czech pre-training ($\rho = 0.03$). This again suggests that a model trained on a mostly projective language can benefit more from further training on non-projective languages than vice versa.

Language	$\lvert S \rvert = 20$ IT	$\lvert S \rvert = 20$ NE	$\lvert S \rvert = 20$ MAML	$\lvert S \rvert = 40$ IT	$\lvert S \rvert = 40$ NE	$\lvert S \rvert = 40$ MAML	$\lvert S \rvert = 80$ IT	$\lvert S \rvert = 80$ NE	$\lvert S \rvert = 80$ MAML
Low-Resource Languages									
Armenian	53.06	**64.61**	63.7	53.47	**64.97**	64.28	54.51	**65.42**	64.84
Breton	57.75	61.36	**64.13**	59.27	62.75	**65.95**	61.95	64.63	**67.48**
Buryat	23.81	26.7	**27.43**	24.46	27.27	**29.27**	25.08	28.22	**30.86**
Faroese	65.26	69.2	**69.39**	66.67	69.85	**70.94**	68.22	70.64	**72.36**
Kazakh	45.69	**55.44**	55.35	46.36	56.08	**56.6**	47.71	57.01	**58.49**
U.Sorbian	50.61	55.46	**56.79**	51.97	56.35	**59.2**	53.25	57.79	**62.3**
Mean	49.36	55.46	56.13	50.37	56.21	57.71	51.79	57.28	59.39
High-Resource Unseen Languages									
Finnish	58.98	**66.77**	66.44	59.31	**67.26**	67.09	59.81	**67.57**	67.44
French	64.12	**66.65**	65.99	64.97	**66.69**	66.26	65.87	66.63	**67.17**
German	74.3	75.69	**76.18**	74.29	75.83	**76.55**	74.34	76.03	**76.9**
Hungarian	58.21	**62.97**	62.87	58.22	**63.43**	61.94	58.54	**63.45**	60.34
Japanese	15.2	40.7	**46.71**	16.35	44.07	**54.38**	18.53	48.88	**60.49**
Persian	46.37	53.67	**54.81**	47.0	54.35	**55.96**	48.25	55.49	**57.93**
Swedish	75.98	**80.85**	80.56	76.22	**80.98**	80.95	76.31	81.09	**81.29**
Tamil	28.86	44.84	**48.1**	30.88	47.47	**53.43**	34.91	50.58	**56.39**
Urdu	19.81	57.05	**57.21**	20.6	57.92	**59.03**	22.29	58.95	**60.69**
Vietnamese	42.7	42.96	**43.94**	42.95	43.48	**44.99**	43.64	44.31	**46.59**
Mean	48.45	59.21	60.28	49.08	60.15	62.06	50.25	61.3	63.52
Mean	48.79	57.82	58.73	49.56	58.67	60.43	50.83	59.79	61.97
Validation & Training Languages									
Bulgarian	76.26	**78.49**	78.04	76.32	**78.53**	78.36	76.48	78.59	**78.7**
Telugu	62.07	69.0	**71.46**	64.09	69.51	**71.37**	65.8	70.74	**73.27**
Arabic	45.03	**71.88**	69.68	48.73	**71.97**	69.94	53.5	**72.01**	70.19
Czech	71.82	**81.14**	80.97	71.93	**81.15**	81.03	72.28	**81.2**	81.12
English	72.04	**81.36**	81.26	72.37	**81.36**	81.32	72.88	**81.39**	81.28
Hindi	29.36	**75.98**	74.24	30.45	**76.06**	74.63	32.49	**76.16**	75.04
Italian	**93.32**	90.61	91.38	**93.32**	90.7	91.55	**93.32**	90.8	91.75
Korean	32.45	**64.79**	63.62	32.86	**65.1**	64.1	34.1	**65.3**	64.37
Norwegian	74.53	**79.28**	78.35	74.67	**79.33**	78.61	74.91	**79.47**	78.79
Russian	71.78	**80.9**	80.06	72.05	**80.96**	80.24	72.41	**81.01**	80.32

Table 12: Results for Italian pre-training. Mean LAS aligned accuracy per support set size $\lvert S \rvert$ for all languages. Best results per category are bolded. Significant results are underlined ($p < 0.005$).

Language	$\|S\| = 20$ CZ	NE	MAML	$\|S\| = 40$ CZ	NE	MAML	$\|S\| = 80$ CZ	NE	MAML
Low-Resource Languages									
Armenian	55.98	**65.01**	64.12	56.75	**65.26**	64.91	58.04	**65.71**	65.66
Breton	52.42	62.44	**64.48**	55.09	63.61	**65.49**	58.23	64.7	**67.02**
Buryat	23.15	27.54	**27.78**	23.71	27.81	**29.44**	24.42	28.86	**31.04**
Faroese	59.19	67.65	**68.24**	60.38	68.89	**70.33**	61.65	69.97	**71.73**
Kazakh	44.61	**55.7**	54.58	45.46	**55.84**	55.49	46.87	56.77	**57.24**
U.Sorbian	56.79	57.0	**58.18**	57.94	57.92	**60.46**	59.53	59.12	**63.68**
Mean	48.69	55.89	56.23	49.89	56.56	57.69	51.46	57.52	59.39
High-Resource Unseen Languages									
Finnish	56.11	**66.15**	65.89	56.46	**66.73**	66.57	56.95	66.84	**67.0**
French	54.69	**65.67**	64.5	55.66	**65.26**	64.63	57.63	**65.37**	65.17
German	65.38	75.3	**75.71**	66.05	75.48	**75.97**	67.56	75.69	**76.42**
Hungarian	52.11	**62.62**	61.11	52.61	**62.76**	60.99	54.74	**63.13**	60.3
Japanese	12.66	40.96	**45.06**	13.84	43.81	**51.61**	16.69	48.47	**57.6**
Persian	50.77	54.23	**55.06**	51.01	55.14	**56.24**	51.72	56.06	**58.29**
Swedish	67.61	**81.04**	79.95	67.88	**81.61**	80.4	68.55	**81.45**	80.89
Tamil	34.53	46.46	**49.42**	36.54	46.94	**54.03**	39.08	52.15	**56.61**
Urdu	22.26	57.54	**57.55**	22.98	58.91	**59.5**	24.44	59.28	**60.98**
Vietnamese	39.52	43.14	**44.35**	40.03	43.72	**45.0**	40.85	44.34	**46.24**
Mean	45.56	59.31	59.86	46.31	60.04	61.49	47.82	61.28	62.95
Mean	46.74	58.03	58.5	47.65	58.73	60.07	49.18	59.87	61.62
Validation & Training Languages									
Bulgarian	76.86	**77.92**	77.32	76.85	**77.95**	77.57	76.84	**78.06**	78.0
Telugu	61.63	68.21	**68.55**	63.28	68.88	**69.97**	65.36	69.45	**71.89**
Arabic	57.77	**72.3**	69.78	58.15	**72.55**	70.17	58.7	**72.31**	70.38
Czech	**90.15**	85.51	85.95	**90.15**	86.34	86.29	**90.15**	85.72	86.65
English	58.9	**80.26**	79.75	60.07	**80.37**	79.89	62.18	**80.4**	79.97
Hindi	30.97	**76.08**	74.09	31.9	**76.21**	74.36	33.83	**76.37**	74.79
Italian	69.39	**85.74**	85.01	70.79	**85.66**	85.06	73.52	**85.72**	85.18
Korean	32.47	**65.4**	63.7	32.89	**66.29**	64.31	33.93	**66.15**	64.63
Norwegian	62.08	**79.47**	77.48	62.58	**79.63**	77.73	63.49	**79.57**	77.89
Russian	76.77	**81.65**	80.34	76.84	**82.17**	80.51	76.95	**81.72**	80.6

Table 13: Results for Czech pre-training. Mean LAS aligned accuracy per support set size $|S|$ for all languages. Best results per category are bolded. Significant results are underlined ($p < 0.005$).