You Can Have Your Data and Balance It Too: Towards Balanced and Efficient Multilingual Models

Tomasz Limisiewicz[♠*†] **Dan Malkin**[◇*] **Gabriel Stanovsky**[◇]
[◇] School of Computer Science, The Hebrew University of Jerusalem
[♠] Faculty of Mathematics and Physics, Charles University in Prague
{dan.malkinhueb,gabriel.stanovsky}@mail.huji.ac.il
limisiewicz@ufal.mff.cuni.cz

Abstract

Multilingual models have been widely used for cross-lingual transfer to low-resource languages. However, the performance on these languages is hindered by their under-representation in the pretraining data. To alleviate this problem, we propose a novel multilingual training technique based on teacher-student knowledge distillation. In this setting, we utilize monolingual teacher models optimized for their language. We use those teachers along with balanced (sub-sampled) data to distill the teachers' knowledge into a single multilingual student. Our method outperforms standard training methods in low-resource languages and retains performance on high-resource languages. [1]

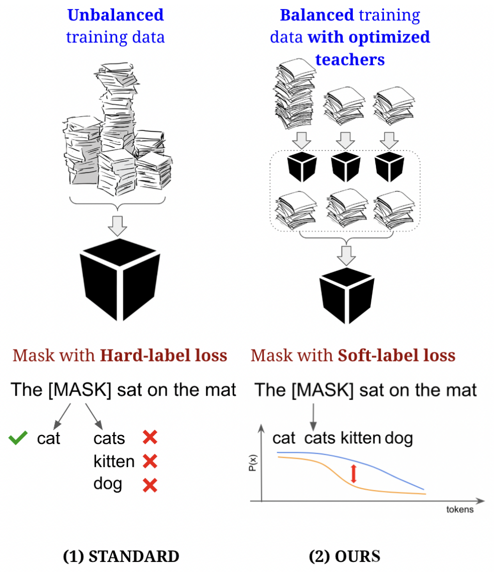

Unbalanced training data
Balanced training data with optimized teachers
Mask with Hard-label loss
Mask with Soft-label loss
The [MASK] sat on the mat
The [MASK] sat on the mat
✓ cat
cats ✗
kitten ✗
dog ✗
cat cats kitten dog
P(x)
tokens
(1) STANDARD
(2) OURS

Figure 1: We train a student for multilingual language modeling using a collection of teachers optimized for each of the target languages and multilingual data subsampled to the data size of the lowest resource language. Our approach achieves a better trade-off in performance between high- and low-resource languages.

1 Introduction

While multilingual language models have been gaining popularity, largely thanks to their cross-lingual transfer ability, their performance has been shown to be skewed toward languages with abundant data (Joshi et al., 2020; Wu and Dredze, 2020). Introducing language models that better incorporate diverse and low-resource languages can increase accessibility to NLP technologies in these languages and help improve cross-lingual transfer (Malkin et al., 2022).

In this work, we address two research questions. First, we ask if we can we improve performance on low-resource languages without hurting it on high-resource ones? Second, does a better trade-off between high- and low-resource languages improve cross-lingual transfer?

To answer these two questions, we distill multiple monolingual teacher models optimized for various languages into a single multilingual student model, using a small balanced multilingual dataset (Figure 1). Our experiments show that this allows taking advantage of data in high-resource languages while avoiding under-fitting low-resource languages.

2 Background: Soft Vs. Hard Labels

We compare two alternatives for the masked LM loss functions: the original loss used for masked language modeling, i.e., *hard labeling* and *soft labeling* as defined in Sanh et al. (2019):
(1) *hard labeling*, which takes into account a single gold masked token in a sentence, y_{gold}, and evaluates the model's prediction for this word, i.e., standard cross-entropy loss:

$$\mathcal{L}_{HARD} = -\log(P(y_{gold})) \tag{1}$$

[*] Equal contribution. The order was decided by a coin toss.

[†] Work done while visiting the Hebrew University.

[1] We will make all of our code and resources publicly available.

(2) *soft labeling*, which allows for multiple valid candidates using the output distribution of an oracle (or a strong LM) $\hat{M_l}$ as a soft label:

$$\mathcal{L}_{SOFT} = -\sum_{y \in V} P_{\hat{M_l}}(y) \log \frac{P(y)}{P_{\hat{M_l}}(y)} \qquad (2)$$

Where y denotes tokens in the model's vocabulary V. Please note that $\mathcal{L}_{SOFT}$ is also equivalent to a KL-divergence between oracle and predicted distributions.

In the following sections, we will explain how soft labeling allows us to distill multiple teachers into a single multilingual student while accounting for balanced performance in high- and low-resource languages.

3 Teacher-Student Distillation for Multilingual Language Models

We train a multilingual student using the masked-language modeling objective and a collection of monolingual teachers optimized for each student's language. All models share one multilingual vocabulary. Sharing vocabulary was necessary to apply our *soft labeling* loss, which requires that the student's and teacher's probability space (in the case of language models: vocabularies) are the same.[2]

To avoid under-fitting low-resource languages, we naively balance the students' training data by truncating data in all target languages to the data size of the lowest resource language. To make the most out of high-resource languages, we rely on soft labeling. For a mask in a given language, we use the *high-resource* language-specific teacher's distribution over the mask and use it as the oracle $\hat{M_l}$ in Equation 2 as a soft label. Our intuition is that this allows the student to gain the broader teachers' knowledge in its language and thus compensate for the sub-sampled data size. Figure 1 provides a visual scheme for this approach.

Formally, given a set of languages $L = \{l_1, l_2, ..., l_K\}$, their corresponding teachers $T_{l_1}, T_{l_2}, ..., T_{l_K}$, and their data $D = \{D_1, D_2, ..., D_K\}$ we teach the student model using the K teachers (which are trained for each of the languages). For student training, we truncate the data size of all languages in D to the smallest dataset size ($min(|D_1|, |D_2|, ..., |D_K|)$).

Size [characters]	Shared Script	Diverse Script
100M	English	Russian
100M	German	German
50M	Spanish	Korean
30M	Hungarian	Greek
20M	Vietnamese	Hindi
10M	Turkish	Telugu
10M	Basque	Urdu

Table 1: Pre-training datasets for each language (in millions of characters) sampled from Wikipedia for high-resource (top) and low-resource (bottom) languages. Some of the selected low-resource languages are actually widely spoken. They were chosen because of relatively smaller Wikipedia sizes (as shown in Appendix B).

Data selection and processing. We collect pre-training data from Wikipedia,[3] aiming to capture a diverse set of high and low-resource languages, as summarized in Table 1. We subsample the corpora by randomly choosing sentences from each language's full corpus. We designate high-resource languages as ones with 50 or 100 million characters in their corpus after sampling, while low-resource languages' corpora consist of 10, 20, and 30 million characters.

Throughout our experiments, we compare 7 languages that share the Latin script versus 7 languages with varying scripts, as the script was found to be an essential factor for multilingual performance (K et al., 2020; Muller et al., 2021; Malkin et al., 2022). We include German in both sets (as one of 7 languages), to compare its performance in both settings.

Models' Architecture and Hyper-parameters. Each of our models comprises of 6 hidden layers and 4 attention heads, an MLM task head. The embedding dimension is 512 and sentences were truncated to 128 tokens. In total, our models consist of 51193168 parameters. We train a single uncased wordpiece tokenizer (Wu et al., 2016) on the 100mb splits of 15 languages.[4] Before tokenization, we strip accents for all languages except Korean.

We train all models for 10 epochs, with a batch size of 8. We used linear decay of the learning rate

[2] Please refer to Section 8, "Teacher model availability" for discussion about vocabulary sharing across monolingual models.

[3] Obtained and cleaned using wikiextractor (Attardi, 2015). We chose Wikipedia as it consists of roughly similar encyclopedic domains across languages and is widely used for training PLMs (Devlin et al., 2019).

[4] 13 languages presented in Table 1 with Hebrew and Lithuanian that were added for future experiments.

with the initial value of 5e-5. Exact configurations and parameters are available in our code.

4 Experiments

We validate our method using two experiments. First, we ascertain that our method indeed improves performance for low-resource languages while maintaining performance for high-resource languages. This is done by comparing the performance of our approach in masked language modeling with two multilingual baselines. Second, we show that our method is competitive for downstream tasks and cross-lingual transfer by probing the pre-trained models for POS and NER tagging.

Multilingual modeling. We evaluate masked language modeling performance on monolingual test sets by measuring *mean reciprocal rank* (MRR). Since the performance of multilingual models is often compared to the performances of monolingual baselines, we report the *average performance difference* between a multilingual model and the monolingual models trained on the same set of respective languages.

Downstream probing. We use the models trained in the previous experiment and train a probe,[5] keeping the base model parameters frozen, to predict part-of-speech tagging (POS) and name entity recognition (NER), as provided respectively by universal dependencies (Nivre et al., 2020) and the XTREME benchmark (Hu et al., 2020).[6] We chose those two tasks because they commonly appear in NLP pipelines (Manning et al., 2014; Honnibal and Montani, 2017). We measure the models' performance in two cases: when the training and test datasets are in the same language (denoted IN-LANG) and when a probe trained for a language l_1 is tested on another one l_2 (denoted ZERO-SHOT). As noted by Hu et al. (2020), zero-shot evaluation is a good measure of a model's cross-lingual transfer. We use probing because it offers a good insight into the representation learned by the model (Belinkov, 2022).

Baselines. We compare the students' performance to multilingual models trained with *hard labels*, on the same data and languages as the student and its teachers. One such model was trained on all the available data in each language to examine the extent of under-fitting low-resource languages, denoted ***HL***. Additionally, to measure how much our student gains from its teacher's knowledge, we train another model on the corpora constrained to the size of the least resourceful language using the standard *hard labels*, denoted ***HL balanced***.

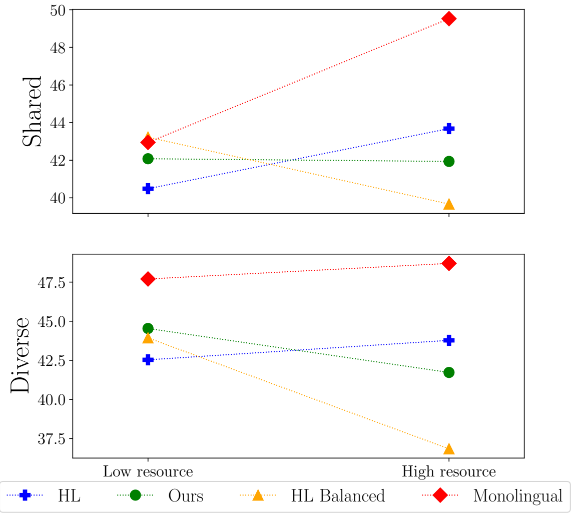


Figure 2: Our balanced teacher-student approach using soft labels presents the overall best combination for low and high-resource languages among multilingual models. This figure presents average MRR results in masked language modeling for both low- and high-resource languages. Results are reported for a Latin-script language set (Shared) and a set with diverse scripts (Diverse).

Experimental Setup Each teacher is a monolingual model trained with *hard labels*. The teachers are trained on the entire training corpus available in their language. In a student model, we distill the knowledge of multiple monolingual teachers into a multilingual student using *soft labels*, as described above. The distillation into the student is performed on groups of shared and diverse script languages. The data is constrained to 10 million characters for each language. All our models are trained using default BERT hyper-parameters detailed in Section 3.

5 Results

We report the experimental results on our test sets, in three language sets grouped by the amount of data available in pre-training, i.e., low-resource, high-resource, and all data. We address our research questions in light of the results:

[5]

[6] See Section D.2 in the Appendix for more information.

Script	Lang. Set	HL	HL Balanced	Ours
Shared	Low-Res.	-2.5	**0.3**	-0.1
	High-Res.	**-5.8**	-10	-7.6
	All	-3.9	-4.0	**-3.7**
Diverse	Low-Res.	-5.1	-3.8	**-3.1**
	High-Res.	**-5.0**	-12	-7.0
	All	-5.0	-7.2	**-4.7**

Table 2: Average difference from monolingual baselines (higher is better) calculated on MRR scores. Our teacher-student model achieves better results overall in both shared and diverse scripts. It is otherwise between the baselines, except for shared script, where it is better for low-resource.

	Lang. set	HL		HL balanced		Ours	
		I-L	Z-S	I-L	Z-S	I-L	Z-S
Shared	Low-Res	35.2	33.4	35.5	34.3	**36.6**	**34.5**
	High-Res	83.3	33.7	81.2	32.4	**84.3**	**33.8**
	{de}	**87.1**	32.3	84.1	32.2	86.8	**33.0**
	All	55.8	33.5	55.1	33.5	**57.0**	**34.2**
Diverse	Low-Res	53.1	35.8	54.6	34.9	**55.7**	**35.9**
	High-Res	76.8	36.2	73.4	34.7	**77.3**	**36.8**
	{de}	**87.7**	36.8	83.3	35.3	87.4	**38.1**
	All	63.3	36.0	62.7	34.8	**64.9**	**36.3**

(a) Accuracy of POS probing.

	Lang. set	HL		HL balanced		Ours	
		I-L	Z-S	I-L	Z-S	I-L	Z-S
Shared	Low-Res	26.5	23.7	27.9	**24.3**	**29.8**	23.9
	High-Res	34.2	24.9	34.7	24.7	**37.6**	**26.0**
	{de}	31.4	**27.4**	**32.1**	25.7	32.0	23.9
	All	29.8	24.2	30.8	24.5	**33.1**	**24.8**
Diverse	Low-Res	25.7	12.8	28.0	**13.8**	**29.9**	12.9
	High-Res	32.8	14.9	29.9	15.1	**37.2**	**17.1**
	{de}	32.5	14.8	31.5	15.7	**35.3**	**17.2**
	All	28.7	13.7	28.8	14.4	**33.0**	**14.7**

(b) Macro F1 of NER probing.

Table 3: For each model and language set, we report average IN-LANG performance (probe trained and tested on the same language) and average ZERO-SHOT performance (probe trained on one language and tested on another). Each ZERO-SHOT number is an average result across all source languages and target languages in the indicated language set. Each entry is a mean of 5 runs with different probe initialization. The Results with significance intervals for each language can be found in Appendix (Tables 6, 7).

Our method offers a good trade-off between performance on high- and low-resource languages. Figure 2 shows the trend of language modeling scores (MRR) when changing from low- to high-resource set. Table 2 summarizes performance differences from monolingual models for our method and the two control baseline models.

In low-resource setting, our model outperforms *HL* and achieves similar results to *HL balanced*. For high-resource languages, our approach closely trails *HL* and is better than *HL balanced*, which was trained on the same data as our student model. It indicates that the student model effectively acquires knowledge from the teachers' distributions. Our model achieves the best results overall when calculated over all languages.

Better trade-off between high- and low-resource languages improves results on downstream. Table 3 shows that IN-LANG and ZERO-SHOT results of probing for POS and NER labels. Our method achieves better or on-par average results in both tasks and language sets. The only exception is *HL balanced* baselines, which scores better in NER for low-resource languages.

Sharing script is not necessary for good multilingual performance. As seen in Figure 2 and Table 2 for low-resource languages, shared script results are consistently closer to monolingual results compared to the diverse script setting. Whereas, for high-resource set, the average difference between the results of monolingual models and our model or *HL* is smaller in the diverse script scenario. For the language included in both sets (German), MRR is higher when coupled with distinct script languages. The performance difference is 0.4 and 0.9 percent in favor of diverse scripts, for *HL* and our model. *HL balanced* scores 2.8% better in shared script scenario. This implies that diverse scripts can benefit multilingual modeling when we reveal enough monolingual data (as in high-resource setting).

In Table 3, we observe that the results for German in the shared-script scenario are better for POS tagging and worse for NER in comparison to diverse-script. Those findings align with previous results suggesting that shared vocabulary is not necessary for cross-lingual transfer and has a varying effect depending on the task (K et al., 2020; Malkin et al., 2022).

6 Related Work

Recent work utilized knowledge distillation in training NLP models. However, to the best of our knowledge, we are the first to do this in low-resource, balanced data settings. Contrary to the approaches of Tsai et al. (2019); Sanh et al. (2019), we do not scale down student models but constraint training datasets.

Sun et al. (2020) use one teacher model and train for machine translation, and Heffernan et al. (2022) use a single multilingual teacher to train a sentence embedding model for low-resource languages. Both rely on parallel corpora for target low-resource languages. Other works on multilingual language modeling addressed how to improve low-resource performance, largely using post-hoc or language-specific solutions. Chau et al. (2020) change the vocabulary to account for low-resource languages, while Muller et al. (2021) transliterate tokens of low-resource languages to the most similar available high-resource language.

Finally, Pfeiffer et al. (2020) introduce cross-lingual adapters, compact components that allow adapting a given model pre-trained for a task in a different desired language.

7 Conclusions

We train multilingual language models aimed at balancing the models' performance for languages with uneven data sizes. We outperform standard models for low-resource languages while maintaining performance on high-resource languages. Noticeably, our method gives better results overall than the naive data sub-sampling. Lastly, our model is a good representation learner for downstream tasks, outperforming baselines for two probing tasks.

Taken together, our results suggest a new direction for multilingual modeling that accounts for a more even performance across low- and high-resource languages and improves cross-lingual transfer.

8 Limitations

Restricted model size and training. Due to limited computational resources, we performed experiments for models significantly smaller than the ones developed by the industry. We based our down-scaling choices on previous ablation studies on cross-lingual models (K et al., 2020). In line with their findings, we prioritized model depth (6 hidden layers) over width (4 attention heads). Also, we examine only BERT based models. This work serves as a proof of concept for a new multilingual language modeling, and future work can extend the study to bigger models with different architectures.

Restricted data. We decided to train our models on sub-sampled Wikipedia to achieve reasonable training times. As shown in appendix B.2 the chosen sample follows the resource-richness trend across languages but does not fully reflect the imbalance between high- and low-resource languages. Nevertheless, we think that this issue does not weaken our point, as even our "unbalanced" baseline model is trained on less skewed data than currently deployed multilingual models. Furthermore, we train our models on 7 languages. Our method needs to be verified on larger data sizes and broader language sets.

Working with limited training data might still be valuable in several aspects. First, there's a growing interest in efficient, and green AI. Smaller and more efficient models will reduce training and inference costs while allowing them to run on less capable hardware and make them accessible to a wider community. Second, from a linguistic perspective, many of the world's languages lack large corpora, and hence will benefit from models that leverage a limited amount of available resources (Joshi et al., 2020).

Naive balancing method. We truncate our training to the size of the smallest low-resource languages, which might be a naive and aggressive approach leading to a sub-optimal performance on our available data. However, our simple approach achieves good results even with naive balancing. Future work can extend it with complex data balancing approaches, such as weighing training data using a learned data scorer (as done in Wang et al. (2020)).

Teacher model availability. Our teacher-student training method assumes the existence of pre-trained monolingual teachers for each considered language, which is considerably less sustainable than training only one multilingual model. Nevertheless, we believe that it is possible to re-use publicly available models as teachers for high-resource languages, while for low-resource languages, competitive results can be obtained with smaller models requiring less computation (Hoffmann et al., 2022). Because our distillation method works on predicted distribution and not latent representations, to combine knowledge of teachers from multiple source languages, we will need to align their vocabularies, which was shown to be feasible by Artetxe et al. (2020); Rust et al. (2021). We leave this engineering task for future work.

Metrics for probing tasks. To evaluate probing for NER we used macro-F1 measured per token and not per entity as in usual NER evaluation. We observed that the probes underperformed in correctly classifying all tokens in a single entity. It led to overall low results in regular F1 that would not allow meaningful comparison between analyzed models. Importantly, macro-F1 equally weights the performance in predicting each class. Thus, it is appropriate to evaluate NER task, where most tokens are annotated as not belonging to any entity.

Acknowledgements

We thank anonymous reviewers for their valuable comments on the previous versions of this article. This work was supported in part by a research gift from the Allen Institute for AI, and a research grant 2336 from the Israeli Ministry of Science and Technology. Tomasz Limisiewicz's visit to the Hebrew University has been supported by grant 338521 of the Charles University Grant Agency and the Mobility Fund of Charles University.

References

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, Online. Association for Computational Linguistics.

Giusepppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, 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.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *arXiv preprint arXiv:2205.12654*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

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.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915, Seattle, United States. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. 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.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online. Association for Computational Linguistics.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. 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 3632–3636, Hong Kong, China. Association for Computational Linguistics.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv preprint*, abs/1609.08144.

Lang.	POS		NER	
	train	test	train	test
de	166849	22458	20000	10000
es	28492	3147	20000	10000
en	21253	5440	20000	10000
eu	5396	1799	10000	10000
hu	910	449	20000	10000
tr	3664	4785	20000	10000
vi	1400	800	20000	10000
ru	67435	11336	20000	10000
ko	27410	4276	20000	10000
el	28152	2809	20000	10000
hi	13304	2684	5000	1000
te	1051	146	1000	1000
ur	4043	535	20000	1000

Table 4: Number of training and testing sentences for POS and NER tasks in XTREME data collection. The data were used to train and evaluate probes on top of analysed models.

A Appendix

In the appendix: we provide details on datasets used in this work Section B; show how proposed teacher-student distillation behaves in the monolingual scenario with just one teacher Section C; present detailed results of our two experimental for each language Section D; provide details of our training procedure and hardware usage Section E.

B Datasets Details

B.1 Data Splits

For pre-training (monolingual) teacher and *HL* models, we use Wikipedia splits of sizes indicated in Table 1, for training student and *HL balanced* models, we subsample training corpus to 10 million characters. We use validation and test sets containing 10000 Wikipedia sentences each.

For downstream probing, we use train and test splits from XTREME. The numbers of sentences in these splits per language are shown in table 4.

B.2 Correspondence of the Sizes of Our Corpora and Wikipedias

Figure 3 shows the per language correspondence between our corpora size and the whole Wikipedia. The latter was used to pre-train MBERT (Devlin et al., 2019). We observe a good linear fit between character numbers in our corpora and the logarithm of Wikipedia byte size. It suggests that the multilingual imbalance is even more severe in the original dataset than in our sample.

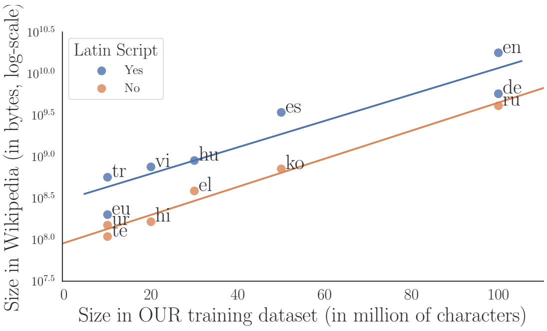


Figure 3: A comparison of subsampled corpora sized and the data available in Wikipedia, which was MBERT's training corpus.

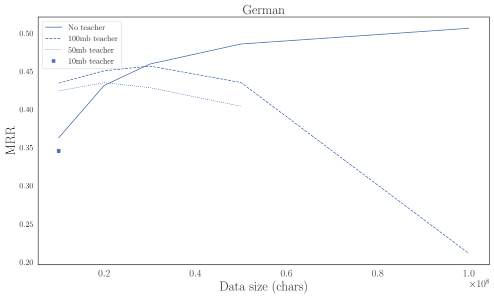


Figure 4: Performance of a language model as the function of training corpora size. The regular HL training is compared with the knowledge distillation to a student on the dataset lower or equal in size than the teacher's training set.

C Teacher-Student Method in the Monolingual World

The purpose of this experiment is to visualize how the model's performance scales with the size of the pre-training dataset. Also, we check the behavior of the teacher-student knowledge distillation with the change of data size used to train a teacher and a student in a monolingual setting.

We train a monolingual model on German Wikipedia data with five sizes (in millions of characters): 10, 20, 30, 50, and 100. Subsequently, we designate 10, 50, and 100 million character models as teachers and distill their knowledge into students on the same size or smaller corpus.[7]

As presented in figure 4, the teacher performance

[7] In monolingual knowledge distillation, we used a learning rate 5 times higher than in the default BERT training script. This choice led to better results.

	Shared script	**Diverse script**
HL	-2.9	**-2.5**
HL Balanced	**-9.2**	-12
Ours	-6.1	**-5.2**

Table 5: Difference from monolingual baseline, for German. German achieves better results in diverse script, except for *HL Balanced*. This suggest that diverse script might help increase language modeling performance.

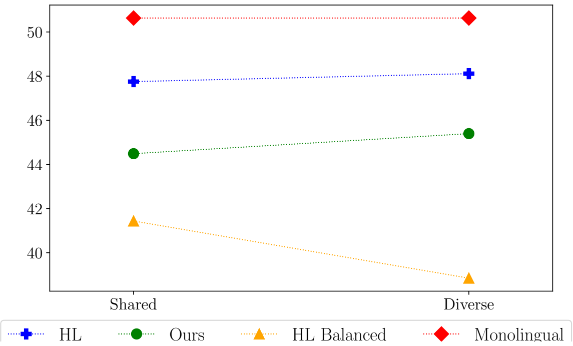


Figure 5: MRR scores for German trained in the set of languages with shared script and diverse script. We observe slight improvement for diverse script over shared script, and significant deterioration for *HL Balanced*.

can be nearly matched by a student trained on a considerably smaller corpus. For the teacher trained on the largest split, the student performance rises steadily with the increase of distillation detest from 10 to 30 million characters and drops after that point. The performance of the student trained on 100 million characters is noticeably low. It is a sign of over-fitting, as in our setting, distillation set is always a subset of the teacher's training set. Also, in the case of teachers trained on smaller corpora, distillation on the dataset of the same size (as the teacher training set) leads to a drop in performance. Therefore, we claim that the distillation is beneficial when the teacher's training set is larger than the student's one.

D Per Language Results

D.1 German: Comparing Shared and Diverse Scripts

Table 5 and Figure 5 present masked language modeling performance for German for three analyzed multilingual model types. German is the language included both in the shared and diverse script language sets. Therefore the results allow comparing which setting is more effective in multilingual language modeling.

D.2 Results for Every Language

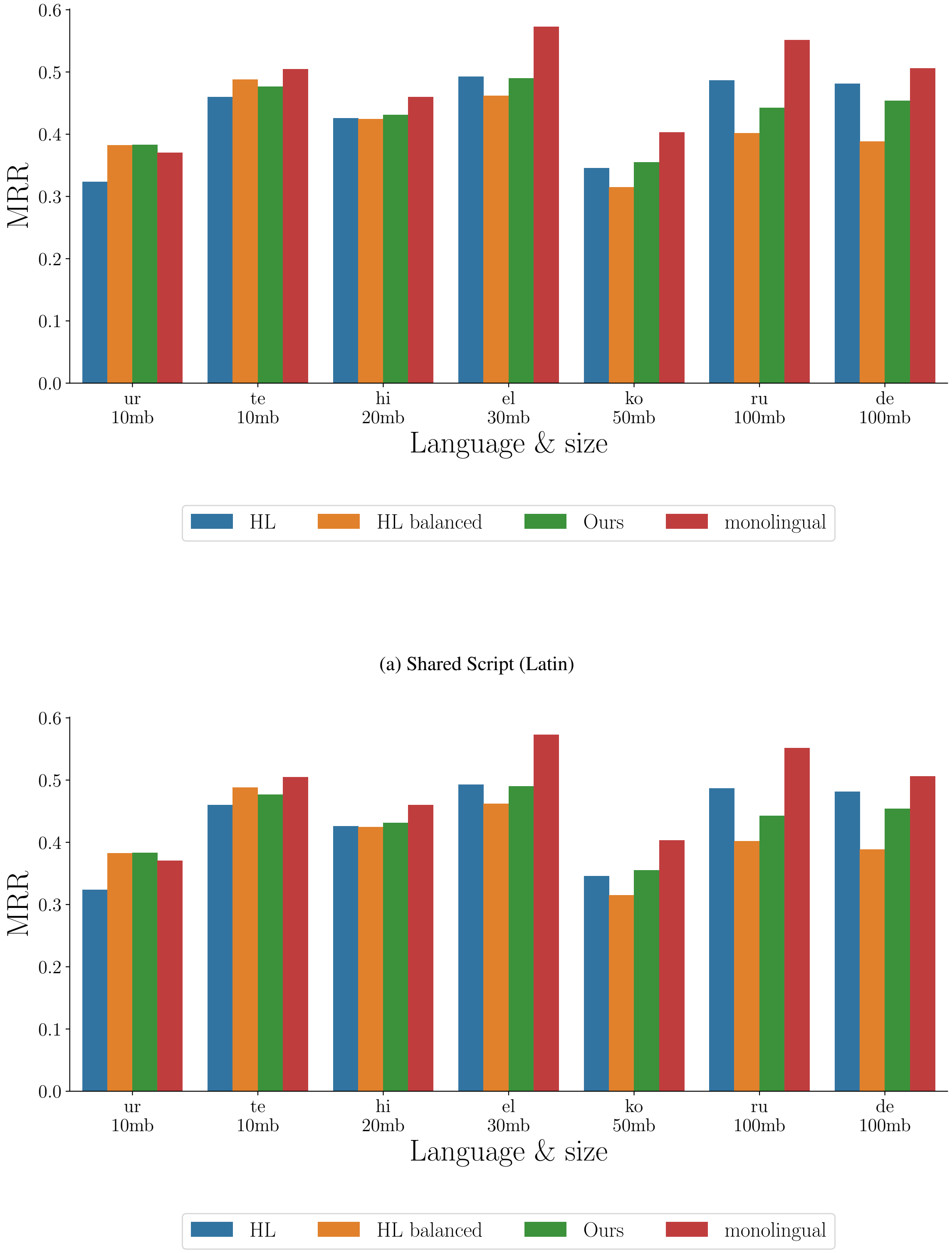
We present per language results in masked language modeling performance in Figure 6 and for probing tasks (POS and NER) in Tables 6 and 7.

E GPUs and training procedures

All of our models (monolingual teachers, students, and multilingual models trained using *hard labels*) are trained on a single GPU core.

We used varying GPUs architectures allocated for each model upon availability (nvidia gtx 980, tesla M60, and RTX 2080Ti). Training time varied between 1 to 3 hours for monolingual models (depending on the data size, language, and GPU core). Multilingual models' training took around 18 hours to complete. Early stopping was used for all models based on results on a balanced dev set.

MLM evaluation was run on the same machines as training or on CPU. the run time ranged from 2 to 4 hours. Training a probe on top of a frozen model took from 1 to 20 minutes, depending on the number of training examples available for a language. The evaluation time on a downstream task was less than 2 minutes.



(a) Shared Script (Latin)

(b) Diverse Script

Figure 6: The figures present MRR results for each language. Our model is compared with baselines: *HL balanced*, *HL* and monolingual models. We observe similar trends as in Figure 2 at higher granularity.

Script	Lang.	HL		HL balanced		Ours	
		In-Lang	Zero-Shot	In-Lang	Zero-Shot	In-Lang	Zero-Shot
Shared	de	87.1 ±0.0	32.3 ±0.9	84.1 ±0.0	32.2 ±1.0	86.8 ±0.0	33.0 ±1.1
	en	79.5 ±0.1	34.2 ±1.4	77.4 ±0.2	32.1 ±2.1	81.1 ±0.2	34.1 ±1.3
	es	83.1 ±0.1	34.6 ±1.7	82.0 ±0.1	32.8 ±1.7	84.8 ±0.1	34.2 ±1.0
	eu	56.3 ±1.2	34.1 ±1.2	58.1 ±1.5	35.0 ±2.7	58.2 ±0.7	33.1 ±2.2
	hu	18.5 ±3.5	37.4 ±1.0	16.6 ±3.4	37.9 ±1.7	18.5 ±5.2	39.5 ±1.2
	tr	40.5 ±2.2	33.3 ±1.5	40.6 ±3.8	34.5 ±2.2	42.1 ±2.6	34.1 ±2.6
	vi	25.5 ±2.2	28.7 ±1.1	26.9 ±3.1	29.9 ±1.3	27.7 ±4.5	31.3 ±1.9
Diverse	de	87.7 ±0.0	36.8 ±0.9	83.3 ±0.0	35.3 ±1.1	87.4 ±0.0	38.1 ±0.3
	ru	79.0 ±0.0	36.9 ±0.9	74.0 ±0.1	36.9 ±1.2	78.6 ±0.1	38.8 ±1.5
	ko	63.7 ±0.2	34.8 ±1.6	62.8 ±0.2	31.9 ±1.8	65.8 ±0.2	33.5 ±1.2
	el	66.6 ±0.2	29.9 ±1.1	66.7 ±0.2	27.0 ±1.0	69.0 ±0.3	30.8 ±1.4
	hi	70.7 ±0.2	34.9 ±0.8	69.6 ±0.1	34.7 ±1.9	70.8 ±0.4	36.0 ±1.6
	te	25.1 ±9.9	42.1 ±1.8	30.0 ±8.4	43.0 ±1.2	28.4 ±6.0	42.6 ±1.3
	ur	50.0 ±1.0	36.3 ±0.6	52.2 ±3.4	34.7 ±0.8	54.4 ±2.0	34.1 ±1.4

Table 6: Accuracy of POS probing for each language. Standard deviations and mean results are computed based on 5 runs with different initialization of the probe.

Script	Lang.	HL		HL balanced		Ours	
		In-Lang	Zero-Shot	In-Lang	Zero-Shot	In-Lang	Zero-Shot
Shared	de	31.4 ±0.6	27.4 ±1.0	32.1 ±0.4	25.7 ±0.4	32.0 ±0.7	26.9 ±1.1
	en	33.0 ±0.5	24.9 ±0.7	33.3 ±0.4	24.8 ±0.2	37.8 ±0.7	25.9 ±0.9
	es	38.2 ±0.6	22.6 ±0.3	38.8 ±1.3	23.7 ±0.7	42.9 ±1.0	25.4 ±1.7
	eu	20.6 ±2.0	27.3 ±0.9	18.5 ±1.3	27.9 ±0.9	20.5 ±0.9	25.9 ±1.0
	hu	26.6 ±0.5	24.3 ±0.9	26.8 ±1.0	25.3 ±0.4	30.2 ±0.6	26.1 ±0.9
	tr	27.3 ±0.5	24.4 ±1.0	30.8 ±0.5	25.4 ±0.4	29.5 ±0.4	24.2 ±0.4
	vi	31.5 ±1.4	18.7 ±0.5	35.5 ±0.5	18.6 ±0.5	39.0 ±1.5	19.2 ±1.3
Diverse	de	32.5 ±0.8	14.8 ±0.6	31.5 ±0.7	15.7 ±0.7	35.3 ±0.4	17.2 ±1.0
	ru	33.7 ±0.8	15.8 ±0.7	29.9 ±0.7	14.6 ±0.7	38.0 ±0.2	16.8 ±0.7
	ko	32.1 ±0.4	14.2 ±0.4	28.2 ±0.5	15.0 ±0.4	38.3 ±0.8	17.3 ±1.1
	el	27.4 ±0.7	16.6 ±0.6	26.5 ±0.9	17.2 ±0.8	31.5 ±0.6	16.6 ±0.6
	hi	16.3 ±0.7	12.8 ±0.4	18.1 ±1.0	14.4 ±1.2	15.7 ±1.1	13.2 ±0.7
	te	13.3 ±1.1	13.8 ±0.5	14.6 ±2.0	13.7 ±0.4	14.2 ±0.6	13.9 ±0.2
	ur	45.6 ±1.1	7.9 ±1.4	52.7 ±1.2	10.0 ±1.2	58.0 ±1.0	8.0 ±0.9

Table 7: Macro-F1 of NER probing for each language. Standard deviations and mean results are computed based on 5 runs with different initialization of the probe.