The SUMEval 2022 Shared Task on Performance Prediction of Multilingual Pre-trained Language Models

Kabir Ahuja[♠] **Antonios Anastasopoulos**[♦] **Barun Patra**[♠]
Graham Neubig[♥] **Monojit Choudhury**[♠] **Sandipan Dandapat**[♠]
Sunayana Sitaram[♠] **Vishrav Chaudhary**[♠]
[♠]Microsoft Corp.
[♦]George Mason University
[♥]Carnegie Mellon University

Abstract

The SUMEval Workshop's shared task involved predicting performance of multilingual PLMs across multiple languages when these models are fine-tuned with varying amounts of data in different languages. The training data was provided for performances of two multilingual models on four NLP tasks, and a baseline was shared with the participants to get started. For test data, the task had two variants for evaluation, *non-surprise* version where the performance was to be predicted for languages seen in the training data but with unseen configurations, and *surprise* version where the languages were unseen during the training. A total of five teams participated in the shared task with 15 submissions overall. The participants proposed addition of new features, feature engineering techniques and trained an ensemble of regression models for the task. The best performing team had an improvement of 64% in MAE over the shared baseline for the *non-surprise* variant, and a 17% improvement for the *surprise* variant.

1 Introduction

Multilingual Pre-trained Language Models (PLMs) (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021; Patra et al., 2022) have been recently gaining prominence due their surprisingly effective cross-lingual transfer capabilities (Pires et al., 2019; Wu and Dredze, 2019). These models are pre-trained on hundreds of languages, and when fine-tuned for a task on a single language (pivot language), they can obtain reasonable performance on languages unseen during fine-tuning (but seen during pre-training). This zero-shot transfer capability while impressive has been found to be non-uniform across languages, and is especially worse on low resource languages or languages that are typologically distant from the pivot language (Wu and Dredze, 2020; Lauscher et al., 2020). Lauscher et al. (2020) showed that these limitations of zero-shot transfer can be addressed by collecting a small amount of data in different languages i.e. the few-shot setup that can substantially improve their performance.

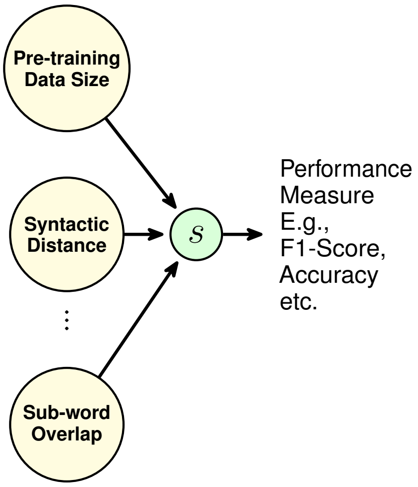

Figure 1: Performance prediction aims to learn a mapping between the factors influencing cross lingual performance of multilingual PLMs like Pre-training Data Size, Typological Relatedness

Despite the fact that these multilingual PLMs support hundreds of languages, most standard multilingual benchmarks (Conneau et al., 2018; Artetxe et al., 2020; Clark et al., 2020; Ponti et al., 2020) support evaluation for only a handful of these, and their performance on a large fraction of languages remain unknown. While creating standardised test sets in all of these supported languages will be an ideal solution, it can be prohibitively expensive to do so.

As pointed out in Ahuja et al. (2022a), performance prediction can be one possible remedy to this problem with multilingual benchmarks, by utilizing the linguistic and model-specific features influencing cross lingual performance to learn a mapping to the observed performance across different languages (See Figure 1). Utilizing regression models for predicting performance on NLP

Proceedings of the First Workshop on Scaling Up Multilingual Evaluation, pages 1–7
Nov 20–23, 2022. ©2022 Association for Computational Linguistics

tasks have been shown to yield meaningful estimates (Xia et al., 2020; Ye et al., 2021), and have also been shown to be effective at predicting performance of multilingual PLMs (Lauscher et al., 2020; Srinivasan et al., 2022; Ahuja et al., 2022b).

The shared task for Scaling Up Multilingual Evaluation (SUMEval) Workshop 2022 entailed this task of performance prediction, where the participants were given the performance of fine-tuned multilingual models XLM-Roberta (Conneau et al., 2020) and the Turing Universal Language Representation model (T-ULRv6) (Patra et al., 2022) for different training configurations across different languages and tasks to build their performance prediction systems. For evaluation there were two versions of the held out test sets, first a *non-surprise* variant where the participants were asked to predict the performance on languages for which some performance data was given in training but with unknown training configurations, and second a *surprise* variant where the performance was to be predicted on languages unseen in the training data.

Participants were provided LITMUS Predictor (Srinivasan et al., 2022) as a baseline to get started and were asked to build better systems possibly using additional features, and alternate prediction algorithms. We saw a participation of five teams for the task, with a total of 15 submissions. Different teams utilized new features in addition to those provided as part of the baseline, alternate feature engineering techniques, and utilized ensemble learning methods for building models. The best performing team on the *non-surprise* variant of the task obtained a 64% reduction in MAE over the baseline, and for *surprise* variant, the best performing team saw an improvement of 17%. To encourage further research in this area we have also made the baseline and datasets available publically[1].

2 Task and Dataset Description

We start by formally defining the performance prediction problem for the shared task. Consider a multilingual model $\mathcal{M}$ pre-trained on a set of $\mathcal{L}$ languages. $\mathcal{M}$ is then to be fine-tuned on some task $\mathfrak{T}$ with labelled data in $\mathcal{P}$ *pivot* languages, and then evaluated on a set of target languages $\mathcal{T}$, where both $\mathcal{P} \subset \mathcal{L}$ and $\mathcal{T} \subset \mathcal{L}$. A training configuration $\mathcal{S}$, is defined by the amount of labelled data for each pivot language $p \in \mathcal{P}$ used for fine-tuning $\mathcal{M}$. The fine-tuned model can then be evaluated on each of the target languages $t \in \mathcal{T}$ to obtain performance measure s, such that s is a function of:

$$s = f(t, \mathcal{S}, \mathcal{P}, \mathcal{M}, \mathfrak{T}) \qquad (1)$$

In performance prediction, the objective is to learn this mapping f, given instances of input configurations $\{t_i, \mathcal{S}_i, \mathcal{P}_i, \mathcal{M}_i, \mathfrak{T}_i\}$ and output performance s_i, so that we can use this mapping to predict performance on unknown training configurations and languages. The input tuple $\{t_i, S_i, \mathcal{P}_i, \mathcal{M}_i, \mathfrak{T}_i\}$ is often represented using various linguistic, model, and data specific features. For a more detailed definition of the task and the features, we refer the readers to Xia et al. (2020); Ahuja et al. (2022a).

In the shared task, we provide the participants different training configurations and their corresponding performance on target languages for 4 multilingual tasks: i) XNLI (Conneau et al., 2018) for Natural Language Inference, ii) TyDiQA (Clark et al., 2020) for Machine Comprehension, iii) WikiANN (Pan et al., 2017) for Named Entity Recognition, and iv) UDPOS (Nivre et al., 2016) for Part Of Speech Tagging; and 2 mulitlingual PLMs: XLM-Roberta (large) and T-ULRv6 (large). The candidates were asked to build regression models using this performance data, and then were evaluated by testing on new training configurations and languages.

2.1 Dataset

The datasets were generated by fine-tuning the models across the 4 datasets along different training configurations and evaluating them on the target languages. The statistics of the datasets are given in Table 1. Training data was released to the participants in the beginning of the competition and the submissions were evaluated on the two variants of the held-out test data:

i) ***non-surprise***: In this test split the participants were asked to predict the performance on the languages for which there was some performance data available in the training set but the training configurations were new, i.e. for the different data allocations of the pivot languages.

ii) ***surprise***: In this test split the participants were asked to predict the performance on new languages, which were unseen in the training dataset (both as a pivot or target language). The training configurations were both new and the ones present in the

[1] https://github.com/microsoft/Litmus/tree/main/SumEval

Task $\mathfrak{T}$	Supp Models $\mathcal{M}$	Dataset Split	Number of Configurations S	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{P}\cap\mathcal{T}\rvert$
XNLI	XLM-R and T-ULRv6	Train	40	15	15	15
		Test (*non-surprise*)	10	15	15	15
		Test (*surprise*)	50	15	10	0
TyDiQA-ID	XLM-R and T-ULRv6	Train	26	9	9	9
		Test (*non-surprise*)	3	9	9	9
TyDiQA-OOD	XLM-R and T-ULRv6	Train	26	9	11	3
		Test (*non-surprise*)	3	9	11	3
WikiANN	XLM-R	Train	400	39	39	39
		Test (*non-surprise*)	100	39	39	39
		Test (*surprise*)	500	39	17	0
UDPOS	XLM-R	Train	400	30	30	30
		Test (*non-surprise*)	100	30	30	30
		Test (*surprise*)	500	30	30	0

Table 1: Dataset statistics for the shared-task. Note that we have 2 versions of TyDiQA: TyDiQA-ID where both training and test set comes from the the original TyDiQA benchmark, and TyDiQA-OOD where the training data is from TyDiQA but test data is from XQUAD (Artetxe et al., 2020).

training data.

For validation, participants were provided scripts for performing Leave-One-Language-Out (LOLO) and Leave-One-Configuration-Out (LOCO) cross-validation from the training data, to help emulate the two test splits. In LOLO, one by one the performance data for each language is kept aside for validation and rest of the data is used for training the model. Similarly, in LOCO each unique configuration is set-aside one at a time for testing and remaining data is used for training.

3 Baseline and Submitted Systems

In this section we will describe the LITMUS predictor baseline and the top two submissions made for the shared task.

3.1 LITMUS Predictor Baseline

The LITMUS Predictor (Srinivasan et al., 2022) is an online open-source tool built to predict task-specific performance of multilingual PLMs across different languages and offering data-collection strategies to improve their performance. The tool utilizes the following features to represent the input tuple $\{t_i, S_i, \mathcal{P}_i, \mathcal{M}_i, \mathfrak{T}_i\}$:

1. Pre-training Data Size of t_i: Cross Lingual performance of multilingual PLMs have been observed to be dependant on the amount of data for a language that was present during pre-training (Hu et al., 2020; Lauscher et al., 2020), where the low resource languages for which the amount of data present in the pre-training corpora was low, are found to benifit less from cross lingual transfer compared to high resource languages. Hence, while predicting the performance for a language t_i we consider the $\log_{10}$ of the size (in tokens) of its pre-training corpus, given by $\texttt{PT-SIZE}(t_i) \in \mathbb{R}$

2. Amount of Fine-Tuning Data in $\mathcal{S}_i$: Fine-tuning multilingual PLMs even with small amounts of labelled data (few-shot-learning) has been found to drastically improve the performance in some cases (Lauscher et al., 2020). Hence, for the given training configuration $\mathcal{S}_i$ representing amount of fine-tuning data in each pivot language in $\mathcal{P}$, we use it as features for the predictor, given as $\texttt{FT-SIZE}(\mathcal{S}_i) \in \mathbb{R}^{|\mathcal{P}|}$.

3. Syntactic Distance between each $p \in \mathcal{P}_i$ and t_i: Target languages that are syntactically closer to the pivot languages have been observed to benefit greater from cross-lingual transfer than the ones that are syntactically distant (Pires et al., 2019; Lauscher et al., 2020). Hence, for predicting performance on t_i, we consider it's syntactic distance with each of the pivot languages $p \in \mathcal{P}_i$, which is computed using the syntactic features provided in the URIEL typological database (Littell et al., 2017). This is denoted as $\texttt{SYN}(\mathcal{P}_i, t_i) \in \mathbb{R}^{|\mathcal{P}|}$

4. Sub-word Overlap between each $p \in \mathcal{P}_i$ and t_i: Finally the sub-word vocabulary overlap between the two pivot and target languages that has also been shown to be important for cross lingual transfer (Wu and Dredze, 2019; Ahuja et al., 2022b) is also considered as a feature, denoted by $\texttt{SWO}(\mathcal{P}_i, t_i) \in \mathbb{R}^{|\mathcal{P}|}$.

These four family of features are then used to represent the input configuration which is used to estimate the performance value:

$$s_i \approx f(\texttt{PT-SIZE}(t_i), \texttt{FT-SIZE}(\mathcal{S}_i), \texttt{SYN}(\mathcal{P}_i, t_i), \texttt{SWO}(\mathcal{P}_i, t_i))$$

f can be approximated using any regression algorithm, and the LITMUS predictor by default uses XGBoost (Chen and Guestrin, 2016), and trains a separate predictor for each task $\mathfrak{T}$ and model $\mathcal{M}$ (which gives 8 predictors for our dataset).

3.2 PICT Team System

The team from Pune Insitute of Computer Technology (Patankar et al., 2022) made submissions for both *non-surprise* and *surprise* variants of the task. They proposed three feature engineering methods for the task in their submission: i) **Multi-Output** : The output of the regression model is expected to be a vector containing performance for each target language in the dataset, inputs are represented by the fine-tuning size of each pivot language; ii) **Single-Output** : Predicting performance of each target language separately, one-hot representations of the languages are appended to the input features; iii) **Single-Output w Language Features** : Apart from the pivot sizes, typological distance features (from URIEL (Littell et al., 2017)) between pivot and target pairs are also appended. The participants train a common model for all the four tasks and the two multilingual models by incorporating one-hot vectors for the two as input features, and encourage cross-task and cross-model transfer. For training the regression models they experiment with CatBoost (Prokhorenkova et al., 2018) and XGBoost.

3.3 GMU Team System

George Mason University team (Akter and Anastasopoulos, 2022) builds on the baseline system by proposing alternate feature engineering techniques and included additional input features for modelling the problem. The participants noted that the feature representation in the existing baseline system added a feature for each pivot language, which may not scale well when different combinations of the fine-tuning languages are used at the test time. They proposed a fixed-size featurization scheme which takes weighted sums of pivot-target overlap features, where the weights are decided by pivot sizes. Additionally, they propose two new features : i) **Presence of Target Language in Pre-training** : A binary feature indicating whether the target language was present during pre-training ; ii) **Target Language Writing Scripts** : A binary vector representing the writing script(s) of the target language obtained from van Esch et al. (2022). Additionally, the GMU team also trained models collectively for all the tasks and MMLMs, and used an ensemble of XGBoost, Multi-Layer Perceptron based regressors for their predictor model.

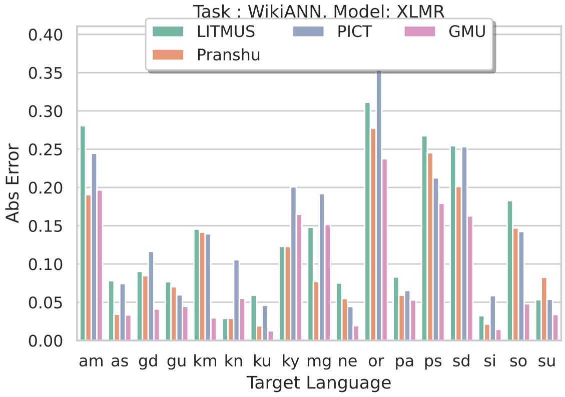

Figure 2: Language wise absolute errors on surprise languages for the baseline and the four submitted systems.

4 Results

We now compare the performance of the submissions and the baseline on both *non-surprise* and *surprise* test sets. Apart from `PICT` and `GMU`, we received submissions from three other teams that we identify by the usernames of the participants i.e. `Khooshrin`, `Viktoria`, and `Pranshu`.

4.1 *Non-Surprise* Test Set

The Mean Absolute Errors (MAE) on the *non-surprise* test set for the baseline and the submissions are given in Table 2. On average, all the submissions out-perform the baseline substantially, with `PICT` obtaining almost 64% reduction in the macro average error (91% in case of micro average). Analysing the task specific errors, we observe the maximum reduction in errors comes from the TyDiQA dataset. This might be attributed to the fact that out of the 4 multilingual tasks, we had the least amount of performance data for TyDiQA (26 training configurations as given in Table 1). Both `PICT` and `GMU` use joint training for multiple tasks which is in contrast to the baseline that trains individual predictors for each task (and model). Hence, the substantial drops in the errors are likely to be

System	Average		TyDiQA		UDPOS	WikiANN	XNLI	
	Macro	Micro	TULRv6	XLMR	XLMR	XLMR	TULRv6	XLMR
LITMUS	0.018	0.131	0.351	0.381	0.005	0.017	0.026	**0.003**
Khooshrin	0.100	0.156	0.301	0.317	0.114	0.085	0.047	0.071
Viktoria	0.030	0.026	0.048	0.037	0.038	0.026	**0.004**	0.004
Pranshu	0.012	0.015	0.019	0.016	**0.006**	0.017	0.026	**0.003**
PICT	**0.011**	**0.011**	**0.012**	**0.014**	0.012	**0.011**	0.008	0.007
GMU	0.023	0.031	0.040	0.054	0.021	0.024	0.032	0.015

Table 2: Mean Absolute Errors (MAE) for the baseline and the submitted systems, on the *non-surprise* version of the test set.

System	Average		UDPOS	WikiANN	XNLI	
	Macro	Micro	XLMR	XLMR	TULRv6	XLMR
LITMUS	0.088	0.055	0.044	0.135	0.025	**0.017**
Khooshrin	0.118	0.070	0.152	0.099	0.016	0.015
Viktoria	0.097	0.064	0.067	0.131	0.028	0.029
Pranshu	0.075	**0.048**	**0.042**	0.109	**0.018**	0.022
PICT	0.104	0.070	0.071	0.141	0.032	0.037
GMU	**0.073**	0.052	0.062	**0.087**	0.026	0.035

Table 3: MAEs for the baseline and the submitted systems, on the *surprise* version of the test set.

attributed to multi-task training which is also in line with the observations in Ahuja et al. (2022b).

4.2 *Surprise* Test Set

Next, we compare the systems on the *surprise* languages test sets in Table 3. Here, teams GMU and Pranshu outperform the baseline with 17% and 14% reduction in macro average errors respectively. Maximum gains are observed for the WikiANN dataset, where GMU team obtains a 35% reduction in MAE. For UDPOS and XNLI tasks, GMU performs slightly worse compared the baseline, while Pranshu obtains comparable errors. We suspect this might be explained by oberving that the errors on WikiANN for the baseline are substantial (±0.135 points F1-Score) compared to the other two tasks, resulting in a better scope for improvement in the former dataset.

We also plot the (surprise) language specific errors on WikiANN dataset for the baseline and the four systems in Figure 2. As can be seen, GMU outperforms the other 4 systems for a majority of the languages, with less then 0.05 error in the F1-score for all languages except Amharic (am), Sindhi (sd), Kyrgyz (kr), Malagasy (mg), Oriya (or), and Pushto (ps) (6 out of 17 languages). This indicates that it might be possible to approximate the performance on new languages with a reasonable accuracy. However, there is still a scope of improvement as the worst case errors are still as high as 0.25 points F1-score for the best performing system.

5 Conclusion

In this paper we presented the findings from the SUMEval workshop shared task on performance prediction of multilingual PLMs. We received 15 submissions from five different teams, and most teams were able to obtain substantial gains over the baseline for the *non-surprise* test set, and two of the teams out-performed the baseline on the *surprise* test set with impressive gains. The strategy of training jointly on multiple tasks and models was utilized by multiple teams, and it lead to substantial improvements for low-resource tasks like TyDiQA. Additional features like the script of the target language were also found to be useful, specially for predicting performance of unseen languages. The best performing system achieved an error of less than 0.05 points F1-score for 11 out of 17 surprise languages for which no performance data was available for training. Overall, the results indicate a promising step towards scaling up the evaluation of multilingual models across multiple languages.

References

Kabir Ahuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2022a. Beyond static models and test sets: Benchmarking the potential of pre-trained models across tasks and languages. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, pages 64–74, Dublin, Ireland. Association for Computational Linguistics.

Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022b. Multi task learning for zero shot performance prediction of multilingual models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5454–5467, Dublin, Ireland. Association for Computational Linguistics.

Syeda Sabrina Akter and Antonios Anastasopoulos. 2022. The GMU System Submission for the SumEval 2022 Shared Task. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. 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.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

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, Valencia, Spain. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Shantanu Patankar, Omkar Gokhale, Onkar Litake, Aditya Mandke, and Dipali Kadam. 2022. To Train or Not to Train: Predicting the Performance of Massively Multilingual Models. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, Online. Association for Computational Linguistics.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022. Beyond english-centric bitexts for better multilingual language representation learning.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

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

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Anirudh Srinivasan, Gauri Kholkar, Rahul Kejriwal, Tanuja Ganu, Sandipan Dandapat, Sunayana Sitaram, Balakrishnan Santhanam, Somak Aditya, Kalika Bali, and Monojit Choudhury. 2022. Litmus predictor: An ai assistant for building reliable, high-performing and fair multilingual nlp systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):13227–13229.

Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara Rivera. 2022. Writing system and speaker metadata for 2,800+ language varieties. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5035–5046, Marseille, France. European Language Resources Association.

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, Hong Kong, China. 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.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Chenchen Ye, Linhai Zhang, Yulan He, Deyu Zhou, and Jie Wu. 2021. Beyond text: Incorporating metadata and label structure for multi-label document classification using heterogeneous graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3162–3171, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.