The GMU System Submission for the SUMEval 2022 Shared Task

Syeda Sabrina Akter and Antonios Anastasopoulos
Department of Computer Science, George Mason University
{sakter6,antonis}@gmu.edu

Abstract

This paper describes the submission of our multilingual NLP model performance evaluation system for the SUMEval 2022 shared task, a system for predict the performance of a model on a set of target languages. The system is based on the LITMUS model (Srinivasan et al., 2022), with the addition of 3 new features and model ensembling. Experimental results show that our system obtains a significant improvement than the baseline on both the test set and the surprised test set. Our system has achieved a 11% MAE reduction on the test set and is the best-performing submission on the surprise test set with 17% MAE reduction compared to the baseline.[1]

1 Introduction

Large multilingual models like mBERT (Devlin et al., 2019), TULRv6 (Microsoft, 2020) and XLM-R (Conneau et al., 2020) are becoming more popular as the foundation of NLP systems that can be used on more than 100 languages. However, most of the languages used in the evaluation of such massively multilingual models are still mostly high-resource ones. A large number of mid- to low-resource languages are not even used as part of the pre-training stage of dearth of unlabeled or labeled data. In addition, training and fine-tuning these large models to evaluate their performance for different combinations of tasks and languages is computationally very expensive. An alternate solution has been provided by making meta-models that can predict performance of multilingual NLP models without running the computationally expensive experiments.

Our submission for the SUMEval 2022 shared task focuses on improving the baseline performance of the LITMUS model by adding different features to the existing ones and using ensembling to improve the performance over the unlabeled test and surprise datasets. Our results show that our method is more effective than the baseline in predicting the performance in the evaluation for an existing as well as unseen set of languages for various settings. In fact, our system ensemble achieves the lowest error for the surprise language test set among the systems submitted to the shared task.

The organization of the rest of the paper is as follows. Section 2 presents the system description of our submitted predictor model. We present evaluation results and perform additional analyses on the SUMEval 2022 datasets in 3 and 4 respectively. We briefly discuss related works in Section 5, and Section 6 presents ideas for further expansion in future work.

2 System Description

Our system is built on top of the baseline LITMUS model (Srinivasan et al., 2022). We first describe briefly this baseline model (§2.1) and then discuss the additional features we use (§2.2). Our best submission consisted of an ensemble of models described in §2.3. From here on, we will be using the terms **target language** to indicate the language on which the fine-tuned model is evaluated (and whose performance we are trying to predict), and **pivot language** to indicate the language on which the model is fine-tuned.

2.1 The LITMUS Model

The LITMUS predictor is an AI assistant for predicting the performance of a multilingual language model like XLMR and mBERT on an NLP task without labeled test data and providing an estimated amount of labeled data needed to achieve the predicted performance for a set of known/unknown languages. The tasks that they focused on are XNLI (Conneau et al., 2018, natural language inference), UDPOS (Silveira et al., 2014, part-of-speech tagging) or WikiANN (Pan et al., 2017,

[1]Our code is available at https://github.com/syedasabrina/GMU_SumEval_litmus.git.

Proceedings of the First Workshop on Scaling Up Multilingual Evaluation, pages 13–20
Nov 20–23, 2022.

named entity recognition). The system introduces a set of factors that may influence zero-shot performance of a multilingual model. These features are largely based on the properties of the models:

Size of Pre-training-data is the $\log_{10}$ of the size of the pre-training corpus per language.

Typological Features capture the similarities based in hand-crafted features inspired by linguistic typology. The typological features for each languages are collected from the WALS database (Dryer and Haspelmath, 2013).

Type overlap with pivot language is a metric that signifies the overlap between the vocabulary of the target language and the vocabulary of the pivot language.

Distance from Pivot Language: is a metric signifying the distance between the target language and the pivot language. This metric is measured using `lang2vec` (Littell et al., 2017) that contains feature vectors for language features such as syntax phonology etc from WALS, SSWL and Ethnologue.

These features are used as an input to an `XGBoost` (Chen and Guestrin, 2016) regressor predictor model after converting them to a [0, 1] range using min-max normalization. The regressor is trained with a learning rate of 0.1, max depth of 10, squared error as the loss function and number estimators of 100 and the error is measured using Mean Absolute Error (MAE).

The evaluation is done under two settings based on the assumption of the availability of labeled test data for a particular target language. The labeled test data is considered unavailable by making the target language not appear in any of the training instances. The MAE of performance predictions across targets has been reported as 0.61%, 0.89% and 0.85% respectively for UDPOS, XNLI and WikiANN when the labeled test data is available and 8.08%, 4.62% and 9.93% when the labeled data is not available.

2.2 Added Features

We have added 3 new features to adapt the LITMUS model to our task-at-hand. Beyond the additional features, we also train the predictor across all tasks and models (as opposed to training a separate model for each task or MLM). The added features are described below:

Fine-tuning Feature: The baseline LITMUS model handles multi-pivot settings by adding $2p$ additional pivot features (capturing pivot-target overlap) for each of the p pivot languages present in the fine-tuning mix. However, under the shared task's settings, one could have different sizes on the combinations of fine-tuning languages with different data sizes used for finetuning. Hence, we decided to introduce a fixed-size "fine-tuning feature" to the LITMUS model, which is calculated using the following equation:

$F = \overrightarrow{L} \cdot \overrightarrow{s}$

where F = Fine-tuning mix feature, $\overrightarrow{L}$ = embedding created for each target language from WALS database (Dryer and Haspelmath, 2013) (similar to the ones already used by LITMUS) and $\overrightarrow{s}$ = data size for each target language to fine-tune the model. Essentially, we compute a single feature, which is the weighted average of the pivot-target overlaps, with the weights being proportional to the amount of data per language in the finetuning mix.

Presence of Target Language in Pre-training: This is a binary feature that indicates if the target language has previously been seen by the model in the pre-training phase.

Target Language Writing Scripts: According to previous works (Muller et al., 2020; Pfeiffer et al., 2020) pre-trained models have been shown to behave differently depending on the language's script. Hence, we have added the information about the writing scripts for the target languages as a feature. For all the languages that are being used in the systems, we have curated a list for all of their writing scripts based on information from van Esch et al. (2022). Note that for each script we have a binary feature depending on the script's usage from each language. Also note that some languages may use multiple scripts, e.g. Hindi both in Devanaghari script and romanized (using Latin script) were used in pre-training of XLM-R. Also note, though, that even though the resource of van Esch et al. (2022) may list multiple scripts for a language, it is not necessarily true that all such data are present in the pre-training or finetuning. We leave such changes to "cleanup" these features for future work.

Training the predictor across all tasks and models: The LITMUS baseline is trained separately for 4 different tasks (UDPOS, WikiANN, XNLI, QA) and for 2 different multilingual models (T-ULR and XLMR). For individual tasks, the predictor is trained on task specific datasets. We have

trained the predictor on all datasets available for all tasks and models combined, using additional categorical features denoting the task and the MLM being modeled.

2.3 Ensemble Learning

Ensembling means combining the predictions from multiple regressors, which in principle should provide better predictive performance. For this task, we have combined the predictions of two different regressors (`XGBoost` and `MLPRegressor`) trained with the additional features on the combined models and combined tasks setting. Though it had a lesser impact on the test set, emsembling significantly improves accuracy on the surprise test set.

3 Evaluation Results

In this section, we are going to analyze the results of the experiments carried out for different test datasets. We have submitted 3 different systems for each test set and the performance is measured on Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). They are the `all_tasks_combined` system, the `all_task_and_models_combined` system, and the ensemble system which are going to be referred as `GMU-Task`, `GMU-Task+Model`, and `GMU-Ensemble` from here onwards. The results for the test set and the surprised set for all 3 of these models are discussed in Sections 3.1 and 3.2 respectively.

3.1 Test Set Results

Based on official leaderboard results, the overall RMSE for the `GMU-Task`, the `GMU-Task+Model` and the `GMU-Ensemble` are 0.016, 0.016 and 0.023 and the MAE is 0.030, 0.030, 0.035. This indicates that the ensemble does not help much in improving the overall accuracy of the system. Training across all models (with the `GMU-Task+Model`) on top of across tasks does not improve over just training across tasks; this indicates that the performance of one model cannot be useful for explaining the performance of another model, at least not using the features we use.

Table 1 provides a breakdown of performance for the systems per task/dataset and language model. Based on the results we make the following observations:

- For UDPOS and WikiANN task, our `GMU-Task` model compares the most with the baseline model. The MAE and RMSE scores are on par between the two models.
- for the XNLI task, the baseline outperforms our best system `GMU-Task` by a large margin, a 78.57% MAE reduction for the XLMR model, and a smaller 13.33% MAE reduction for the T-ULR model over our system.
- For the QA task, our models significantly outperforms the baseline for both XLMR and T-ULR models. The `GMU-Ensemble` System has an 85.3% MAE improvement for the XLMR model over the baseline. Our `GMU-Task` System has a 93.62% of MAE improvement over the baseline for the T-ULR model.
- On average, our `GMU-Task` system has achieved a 76.47% MAE reduction for T-ULR model over the baseline while being on par with the average MAE of the baseline for the XLMR model. Hence, the `GMU-Task` system has an overall MAE reduction of 11% over baseline. Our improvement is attributed to the large improvement on the QA task.

Amongst all our submitted models, the `GMU-Task` system has the better performance numerically. However, `GMU-Task+Model` system is not very far off. These two models have very similar values of MAE and RMSE across tasks and models. The `GMU-Ensemble` has shown the best performance for the QA task for the XLMR but due to its comparatively poor performance over the other tasks, its average performance is poor amongst all systems.

3.2 Surprise Test Set Results

A surprise test set was available for the UDPOS, WikiANN, and XNLI tasks. Based on public leaderboard results, the overall RMSE for the `GMU-Task`, the `GMU-Task+Model` and the `GMU-Ensemble` are 0.10, 0.099 and 0.099, with the MAE at 0.080, 0.082, 0.073. This indicates that emsembling can raise the overall accuracy of the system on unseen languages and settings.

Table 2 presents a score breakdown as before. We summarize some interesting observations below:

- Similar to the test set, for the UDPOS task baseline outperforms our best system by 40.9% MAE reduction.

Model	UDPOS MAE	UDPOS RMSE	WikiANN MAE	WikiANN RMSE	XNLI MAE	XNLI RMSE	QA MAE	QA RMSE	Average MAE	Average RMSE
Baseline (as provided by the organizers)										
XLMR	**0.005**	**0.009**	**0.017**	**0.033**	**0.003**	**0.004**	0.375	0.376	**0.015**	0.043
T-ULR	-	-	-	-	**0.026**	**0.037**	0.345	0.349	0.119	0.194
Combining All Tasks: GMU-Task										
XLMR	0.009	0.014	0.019	0.034	0.014	0.029	0.104	0.130	**0.015**	**0.030**
T-ULR	-	-	-	-	0.030	0.040	**0.022**	**0.034**	**0.028**	**0.038**
Combining All Tasks+Models: GMU-Task+Model										
XLMR	0.009	0.013	0.020	0.035	0.017	0.032	0.081	0.098	0.016	0.029
T-ULR	-	-	-	-	0.030	0.041	0.037	0.051	0.032	0.044
Ensembling: GMU-Ensemble										
XLMR	0.021	0.032	0.024	0.037	0.015	0.021	**0.055**	**0.068**	0.023	0.035
T-ULR	-	-	-	-	0.032	0.042	0.042	0.050	0.034	0.045

Table 1: Results on the test data. Our models significantly outperform the baselines for the QA task (both models), and perform largely on par for almost all other tasks and models. We **highlight** the best performing model per task.

Model	UDPOS MAE	UDPOS RMSE	WikiANN MAE	WikiANN RMSE	XNLI MAE	XNLI RMSE	Average MAE	Average RMSE
Baseline (as provided by the organizers)								
XLMR	**0.044**	**0.059**	0.135	0.164	**0.017**	**0.020**	0.090	0.124
T-ULR	-	-	-	-	**0.026**	**0.028**	**0.025**	**0.027**
Combining All Tasks: GMU-Task								
XLMR	0.070	0.088	0.090	0.116	0.062	0.080	0.080	0.101
T-ULR	-	-	-	-	0.083	0.099	0.079	0.094
Combining All Tasks+Models: GMU-Task+Model								
XLMR	0.082	0.101	**0.080**	**0.099**	0.057	0.072	0.083	**0.100**
T-ULR	-	-	-	-	0.050	0.060	0.055	0.066
Emsembling: GMU-Ensemble								
XLMR	0.062	0.081	0.086	0.115	0.034	0.040	**0.074**	**0.100**
T-ULR	-	-	-	-	0.027	0.033	0.026	0.032

Table 2: Results on the Surprise Test dataset. The baseline model is the best for the UDPOS and XNLI tasks, while out GMU-Task+Model is the best for WikiANN. Note though that our GMU-Ensemble is the best *general* solution, performing the best across tasks on average.

- For the WikiANN task, our GMU-Task+Model has a 40.74% of MAE reduction over the baseline.
- The baseline has an improvement of 50% over our GMU-Ensemble for the XNLI task and XLMR model. However, for the T-ULR model, the performance are on par with each other.
- On average, the GMU-Ensemble has achieved an overall 17% MAE reduction over baseline making it the best general solution, performing the best across tasks on average. This improvement can be attributed to the average performance of the system being greater than the baseline for the XLMR model and being on the same level for the T-ULR model.

Amongst the three submitted systems, the GMU-Ensemble performs the best for the given dataset for the UDPOS and XNLI tasks for both XLMR and T-ULR model. However, the GMU-Task+Model outperforms the GMU-Ensemble for the WikiANN task. From the discussion above we can conclude that ensembling predictions technique can better the performance of the LITMUS model for the surprise test set.

4 Analysis

We have performed various analyses, choosing to focus on the surprise test set and the performance of our best-performing GMU-Ensemble model.

Task	Model	Config	Lang	MAE
UDPOS	XLMR	Diff	Galician	0.033
UDPOS	XLMR	**Same**	Galician	**0.026**
WikiANN	XLMR	Diff	Gujarati	0.027
WikiANN	XLMR	**Same**	Gujarati	**0.020**
XNLI	T-ULR	Diff	Bengali	0.017
XNLI	T-ULR	**Same**	Marathi	**0.016**
XNLI	XLMR	Diff	Panjabi	0.015
XNLI	XLMR	**Same**	Panjabi	**0.013**

Table 3: Lowest MAE's for languages across tasks. We can see for XLMR model, we get the lowest MAE's for the same languages across tasks regardless of configurations. Also, the same configuration data has better performance(highlighted) as the languages are seen by the models during pretraining.

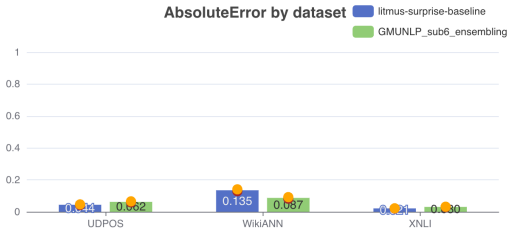


Figure 1: MAE per task for surprise dataset. The `GMU-Ensemble` performs better overall. They have similar trend of values across tasks.

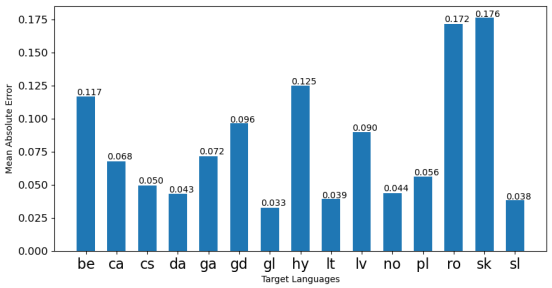
Semantic vs Syntactic Tasks According to the authors (Srinivasan et al., 2022), for the baseline LITMUS model, the predictor relies mostly on the pretraining data size feature for the semantic task and on the typological features and overlap between the language feature for the syntactic task. Figure 1 presents the MAE per task for the baseline and the `GMU-Ensemble` system. The baseline model has the best MAE score for the XNLI task, which is a semantic task and the other 2 tasks which are both syntactic tasks, has poor MAE compared to XNLI. The same trend of the MAE score is also followed by the `GMU-Ensemble` system attributing to the similar feature importance concept of the baseline model.

Performance per Language We also observe large variability in the performance of the `GMU-Ensemble` model across languages and datasets. Figure 2 presents a breakdown of the MAE on surprise tests per target language. The Oriya language has the highest MAE for the WikiANN task. Amongst all the tasks the XNLI task has the lowest MAE for the Panjabi language. The error ranges from 0.013 to 0.246.

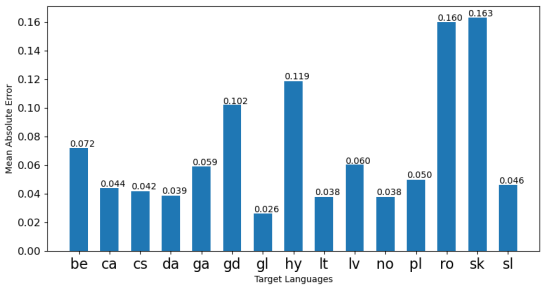
Table 3 catalogs the languages with the lowest MAEs for each tasks and their configurations. Test configurations that are the same as the ones seen during training for each task almost always lead to lower MAE value than the test sets containing surprise languages as well as new configurations. The values reflect the common observation that the models will perform better if the target language has been seen by the model in the pre-training and/or finetuning step. his may also provide a supporting argument for works that promote an equitable allocation of data labeling across languages for multilingual models, e.g. Debnath et al. (2021). Also, it should be noted that regardless of the configuration, we obtain the lowest MAEs for the same languages per task (e.g. for UDPOS with XLMR the lowest MAE under both seen and unseen configurations is for Galician). The only exception is the combination of XNLI with the T-ULR model for Bengali and Marathi. For the XNLI T-ULR Same Configuration test set, we get 0.037 and 0.016 MAE scores and for the XNLI T-ULR Different Configuration test set we get 0.012 and 0.039 MAE scores for Bengali and Marathi respectively. The values are completely reverse of each other. This trend is consistent with other languages for these two datasets. The languages we get the lowest values for the XNLI T-ULR Same Configuration test set are the ones for which we get the highest values for the XNLI T-ULR Different Configuration test set, which is an interesting observation.

5 Related Work

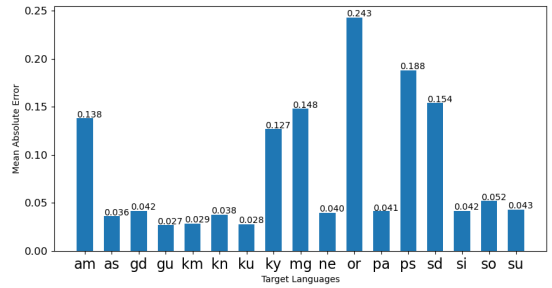
Lin et al. (2019) first explored how to determine which high-resource transfer language can be used to maximize performance in a lower-resource target language in a traditional cross-lingual transfer learning scenario. Given the experimental settings as input, Xia et al. (2020) introduced the performance prediction task for simple cross-lingual transfer settings, constructing regression models that are similar to our system to predict the evaluation outcome of an NLP experiment. Experimenting on nine different NLP tasks, the study discovered that the predictors can make meaningful predictions over unknown languages and different modeling architectures, outperforming baselines and human expert predictions. Ye et al. (2021) then discussed how the task of estimating a system's performance without running the computationally



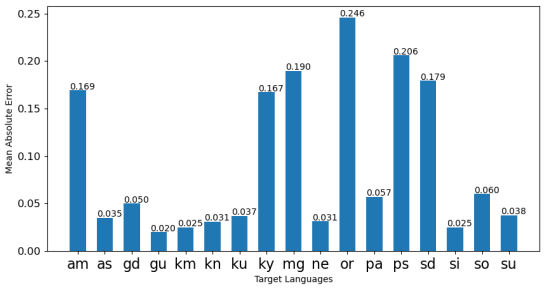
(a) UDPOS XLMR surprise langs diff config



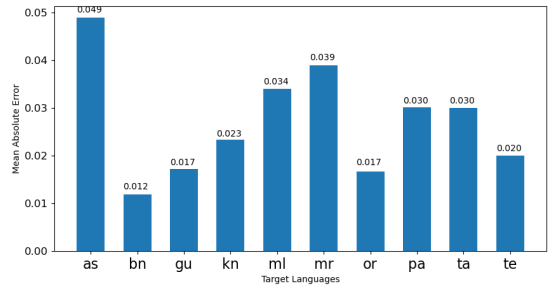
(b) UDPOS XLMR surprise langs same config



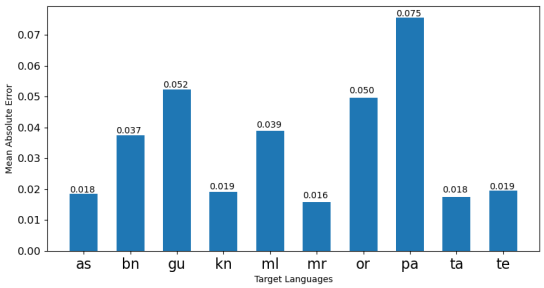
(c) WikiANN XLMR surprise langs diff config



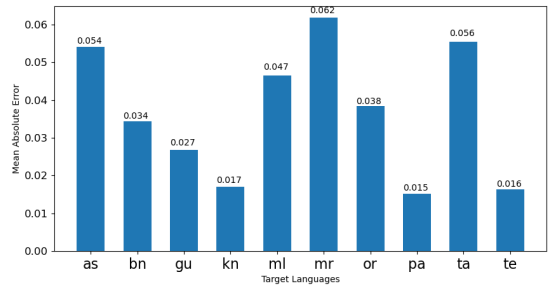
(d) WikiANN XLMR surprise langs same config



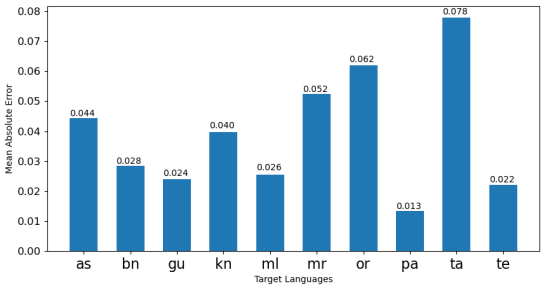
(e) XNLI TULRv6Large surprise langs diff config



(f) XNLI TULRv6Large surprise langs same config



(g) XNLI XLMR surprise langs diff config



(h) XNLI XLMR surprise langs same config

Figure 2: Mean Absolute Errors for the surprise test set broken down by target language. Same languages have lower MAEs across tasks. The XNLI T-ULR dataset has the lowest MAEs for different Languages.

expensive experiments may aid in estimating performance of a language model for new datasets/languages. The study explores approaches for the reliability analysis of performance prediction models after examining the effectiveness of several such performance prediction models on four common NLP tasks.

6 Conclusion

In this paper we describe the GMU team submission for SUMEval 2022 shared task. Our system has extended the LITMUS model by including new features, combining data for training and using ensembling techniques that has improved the overall predictions for the test set.

Acknowledgements

This work was generously supported by NSF Awards FAI-2040926 and BCS-2109578.

References

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*, pages 785–794.

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.

Arnab Debnath, Navid Rajabi, Fardina Fathmiul Alam, and Antonios Anastasopoulos. 2021. Towards more equitable question answering systems: How much more data do you need? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 621–629, 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.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. 2019. Choosing transfer languages for cross-lingual learning. *arXiv preprint arXiv:1905.12688*.

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.

Microsoft. 2020. Turing-nlg: A 17-billion-parameter language model by microsoft.

Benjamin Muller, Antonis Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2020. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. *arXiv preprint arXiv:2010.12858*.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. UNKs everywhere: Adapting multilingual language models to new scripts. *arXiv preprint arXiv:2012.15562*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

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. In *Thirty-sixth AAAI Conference on Artificial Intelligence. AAAI. System Demonstration*.

Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara E Rivera. 2022. Writing system and speaker metadata for 2,800+ language varieties. In *Proceedings of LREC*.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. *arXiv preprint arXiv:2005.00870*.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable nlp performance prediction. *arXiv preprint arXiv:2102.05486*.