To Train or Not to Train: Predicting the Performance of Massively Multilingual Language Models

Shantanu Patankar[1*], **Omkar Gokhale**[1*], **Onkar Litake**[2], **Aditya Mandke**[2], **Dipali Kadam**[1]
Pune Institute of Computer Technology, Pune, India[1]
University of California, San Diego[2]
shantanupatankar2001@gmail.com, omkargokhale2001@gmail.com,
olitake@ucsd.edu, amandke@ucsd.edu, ddkadam@pict.edu

Abstract

Evaluating the performance of Massively Multilingual Language Models (MMLMs) is difficult due to the shortage of evaluation datasets in low-resource languages. Due to computational limitations evaluating MMLMs trained on all possible pivot configurations is not feasible. This paper describes our contribution to the SumEval 2022 shared task, which handles the crucial task of Performance prediction of MMLMs. We build upon Microsoft Research's Project LITMUS and devise a method to further improve predictions. We develop various machine-learning approaches which outperform the baseline score provided by LITMUS. Our system ranked first with an RMSE score of 0.017 for the non-surprise and 0.109 for the surprise dataset.

1 Introduction

Massively Multilingual Language Models (MMLMs) are models that are pre-trained on a large set of languages and can perform various tasks. For example, a Massively Multilingual Neural Machine Translation model (Arivazhagan et al., 2019) is a single model trained on 100+ languages with over 50 billion parameters. Such pre-trained models work very well for zero-shot transfer across languages. However, the performance of these models is not consistent for all languages. They depend on factors like the pivot languages used for fine-tuning and the number of data points used for training. It is not feasible to evaluate the performance of the MMLMs on all languages. This is because some target languages are low-resource and lack proper evaluation sets for testing the performance. It is also difficult to train and test the models on all combinations of tasks, pivot languages, and target languages. This paper aims to develop a system that will take parameters like the MMLM model, task name, pivot languages, and the number of data points used for fine-tuning to predict the model's performance for the task on a particular target language. We develop two different systems. The first is for models fine-tuned on specific pivot languages and then tested on the same target languages. The second system is for models fine-tuned on a set of pivot languages and tested on surprise languages that were not part of the aforementioned set of pivot languages.

[*]first author, equal contribution

2 Related Work

Previously researchers have explored predicting the performance of machine learning models from unlabeled data by utilizing underlying information about data distribution (Domhan et al., 2015) or by measuring (dis)agreements between multiple classifiers (Platanios et al., 2014).

As the NLP Models are getting computationally complex to train, researchers have been interested in predicting the performance of NLP models without actually training them. Xia et al. (2020) have used ten different language features to train a XGBoost regressor. They compare the model's performance with predictions made by human experts. Dolicki and Spanakis (2021) leverage various syntactic features to implement a zero-shot performance predictor. Ahuja et al. (2022) demonstrate a single-task and multi-task performance prediction and discuss the significance of various linguistic features. Srinivasan et al. (2022) have developed LITMUS, a tool for prediction and labeling plan generation. We use LITMUS as a baseline for evaluating the performance of our system. We build upon all these past works by utilizing the syntactic features and tree-based models that have produced good results in the past and implement them on different configurations of data.

3 Dataset Description

The dataset consists of performance measures of XMLR and TULRv6Large, which are finetuned on

Proceedings of the First Workshop on Scaling Up Multilingual Evaluation, pages 8–12

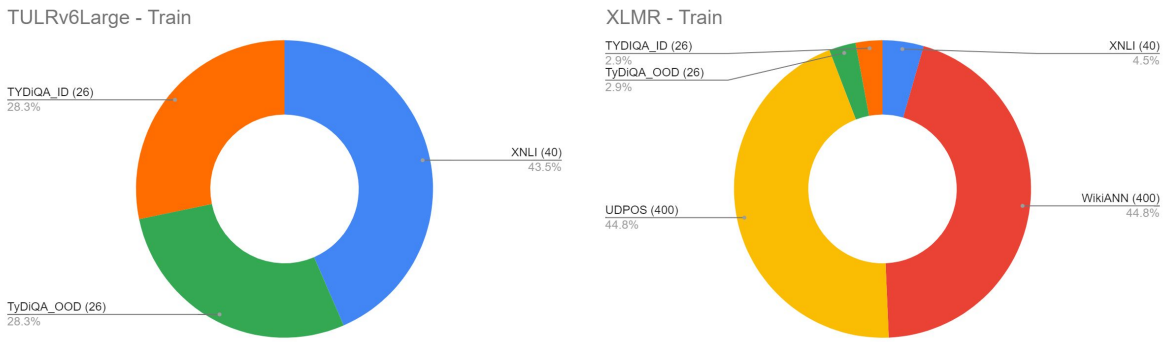

Figure 1: Data distribution of train data.

a specific set of languages (pivot languages) for four different tasks: XNLI (Conneau et al., 2018), WikiANN (Pan et al., 2017), UDPOS, and TyDIQA (Clark et al., 2020)). The dataset has 880 data points with distribution as shown in figure 1. Each data point has the model's training configuration, including the model name, task name, pivot languages, and evaluation results on specific target languages. The training configuration also contains the data used in each pivot language to finetune the MMLM. The evaluation results consist of the performance of the corresponding model on a set of languages. Instead of training a new regressor for every model-task pair, we observed that combining the data helps the prediction model gain better insights. Our four data combination techniques are described as follows:

- **Multi Output Dataset**: The total number of unique languages across all the individual datasets is 40. In order to combine the individual task-model pair-wise datasets, we create 40 columns each for training configuration and evaluation results (one column for each language). A zero in a pivot language column indicates the absence of that language while finetuning. We use this dataset to train a multi-output regressor.

- **Single Output Dataset**: We create a new row for each new evaluation language and provide the target language as an extra feature. We then use this dataset to train a single-output regressor.

- **Single Output Dataset with Language features**: We create an additional dataset by adding a few language features to it. We obtain the pair-wise genetic, syntactic, phonetic, geographic, inventory, and featural distances between target and pivot languages and utilize them as features for the model. These distances are calculated using the URIEL typological database. (Littell et al., 2017).

- **Surprise Dataset**: To predict the performance of MMLMs on surprise languages, we calculate the pair-wise syntactic, phonetic, featural, inventory, genetic, and geographic distances and the subword overlap between the target surprise language and the pivot languages. The target surprise language is also taken as a feature, but we encode the surprise languages with integers that are not present in label encodings of the pivot languages.

Model	Dataset Name	MAE	RMSE
XG-Boost	Multiouput	**0.007**	0.030
	Single output	0.015	0.052
	Single output feats	0.012	0.041
Cat-Boost	Multiouput	0.017	0.035
	Single output	0.012	0.034
	Single output feats	0.008	**0.017**
Litmus	Non-surprise	0.018	0.054

Table 1: Results for non-surprise data.

Model	Dataset Name	MAE	RMSE
XG-Boost	Surprise	0.093	0.128
Cat-Boost	Surprise	**0.082**	**0.109**
Litmus	Surprise	0.088	0.122

Table 2: Results for surprise data.

4 System Description

To get the relationship between different languages, we use different parameters used by Lin et al. (2019) like syntactic, phonetic, featural, inventory,

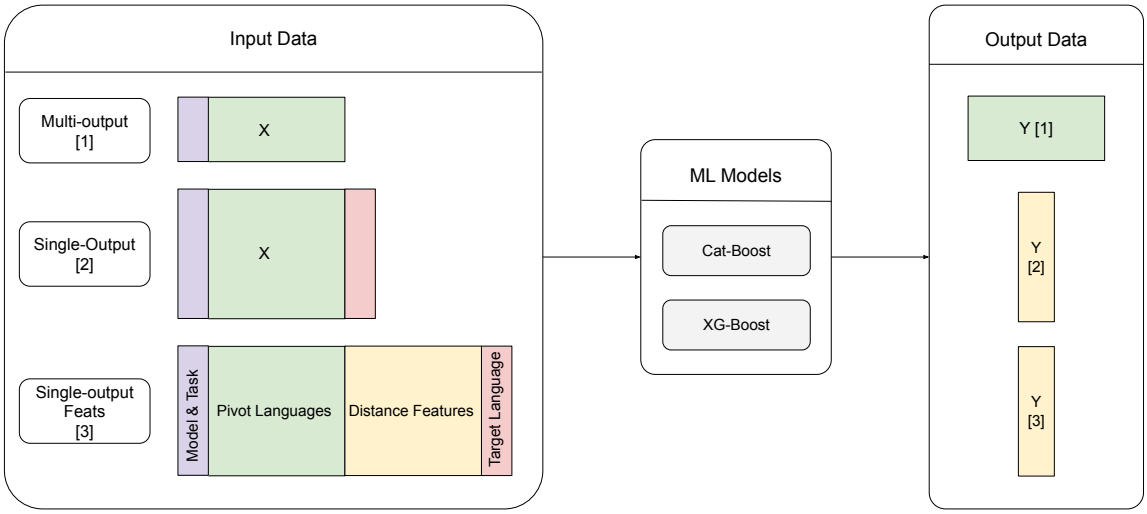

Figure 2: System Design.

genetic and geographic distances, and subword overlap.

- **Syntactic Distance:** The cosine distance between the feature vectors derived from the syntactic structures of the languages.
- **Genetic Distance:** The genealogical distance of the languages.
- **Geographic Distance:** The orthodromic distance between the languages, divided by the antipodal distance.
- **Inventory Distance:** The cosine distance between the phonological feature vectors derived from the PHOIBLE database.
- **Phonological Distance:** The cosine distance between the phonological feature vectors derived from the WALS and Ethnologue databases.
- **Featural Distance:** Cosine of all distances mentioned above.
- **Subword Overlap:** Percentage of common tokens in both languages

We have made two separate systems for performance prediction. The first one is for predicting the performance of the MMLMs on known languages, as shown in Figure 2. The second one is to for predicting the performance of the MMLM on surprise languages.

4.1 Non-Surprise system

We use the three datasets mentioned in section 3 to predict the performance metric of an MMLM on a target language.

4.1.1 Multi Output Model

The dataset has 42 features, 40 denoting the number of data points of the pivot languages, one feature for the model name, and one for the task name. Our targets are the evaluation scores of 40 target languages. We train different regression models like CatBoost (Prokhorenkova et al., 2018), XGBoost (Chen and Guestrin, 2016) and SVM as multi-target regression models on this data.

4.1.2 Single Output Model

The dataset has 43 features, 40 denoting the number of data points of the pivot languages, one feature for the model name, one for the task name, and one representing the target language. Our target is the evaluation score of an individual target language. This dataset is used to train XGBoost, CatBoost, and SVM regressors.

4.1.3 Single output with features model

The dataset has 283 features, 40 for the data size of each pivot language used for fine-tuning, and 240 are the pair-wise syntactic, phonetic, genetic, geographic, inventory, and featural distances of the target with the pivot language. The rest of the features are the model name, task name, and the name of the target language. We train the aforementioned three regressors on this dataset.

4.2 Surprise system

We use the Surprise dataset to train this system. As mentioned above in section 3, this dataset consists of the syntactic, phonetic, featural, inventory, genetic, and geographic distances and the subword overlap of the surprise languages with the pivot languages. The final training data consists of 563 features. 70 features are pivot languages, 490 are the 7 distance parameters of each of the 70 languages with the target surprise language, and the remaining three are for the model name, task name, and target language name. We train CatBoost with a maximum tree depth of 7 and a learning rate of 0.3. For XGBoost, we obtained the best results using the default parameters.

5 Experiments and Results

Our Training setup was pretty straightforward. Some of the observations we made during our extensive experimentation are as follows.

1. **Linguistic features improve the performance**:
   We observed that adding the seven linguistic features mentioned in section 4 improves the score of both the single output regressors. Adding pairwise linguistic features in multi-output data sets is not feasible as we need to add 11,200 new columns.

2. **Tree based models perform better**:
   We tried various regression models such as Logistic Regression, SVM, Multi-Layer Perceptron, Polynomial Regression, Lasso Regression, XGBoost, and CatBoost. We observe that XG-Boost and Cat-Boost are the top-performing models. We speculate this because tree-based machine learning models are good at handling complex, non-linear relationships.

3. **Target language: anonymous vs labeled**:
   When trained on a single output dataset, if we remove the labels of the target language, we observe a consistent but slight reduction in performance. This shows that the model makes informed choices based on the target language.

4. **Dataset: individual vs combined**:
   The model trained on the combined dataset produces better results than training individual models for Task-model pairs. This indicates that the insights gained by a model on a task are transferable.

5. **Features: PCA and Feature Elimination**:
   Performing Principal Component Analysis (PCA) on the extracted features reduces the performance of the models. This indicates that some important features are lost during the decomposition process. Feature elimination does not improve the model performance either.

6. **Eliminating individual language features**:
   We retrain each model by eliminating one syntactic feature and evaluate its performance. We find that eliminating any feature gives a lower overall score than we get by utilizing all the features. We also find that the importance of each feature from most important to least important is as follows:
   1. phonological distance
   2. inventory distance
   3. featural distance
   4. genetic distance
   5. syntactic distance
   6. geographic distance

6 Conclusion

In this paper, we have developed two approaches for the performance prediction of Massively Multilingual Models. One is for known languages, and another is for unknown or surprise languages. We have performed feature engineering on the data using different methods and tested different regression models on these features. For the non-surprise system, CatBoost gave the best performance on the single-output dataset with language features. On the surprise system, too, CatBoost outperformed all the other models. Both systems were able to outperform the LITMUS model. The system's performance can be further improved if more data is available for certain tasks like TyDiQA and XNLI.

References

Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022. Multi task learning for zero shot performance prediction of multilingual models. *arXiv preprint arXiv:2205.06130*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

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.

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, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Błażej Dolicki and Gerasimos Spanakis. 2021. Analysing the impact of linguistic features on cross-lingual transfer. *arXiv preprint arXiv:2105.05975*.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-fourth international joint conference on artificial intelligence*.

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. Association for Computational Linguistics.

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.

Emmanouil Antonios Platanios, Avrim Blum, and Tom M Mitchell. 2014. Estimating accuracy from unlabeled data. In *UAI*, volume 14, page 10.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

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*.

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