LITMUS Predictor: An AI Assistant for Building Reliable, High-Performing and Fair Multilingual NLP Systems

Anirudh Srinivasan[*†1], **Gauri Kholkar**[*2], **Rahul Kejriwal**[*2], **Tanuja Ganu**[3], **Sandipan Dandapat**[2], **Sunayana Sitaram**[3], **Balakrishnan Santhanam**[2], **Somak Aditya**[3], **Kalika Bali**[3], **Monojit Choudhury**[3]

[1] The University of Texas at Austin
[2] Microsoft India Development Center
[3] Microsoft Research Lab India
anirudhsriniv@gmail.com, {gakholka,rakejriw,taganu,sadandap,susitara,basantha,t-soadit,kalikab,monojitc}@microsoft.com

Abstract

Pre-trained multilingual language models are gaining popularity due to their cross-lingual zero-shot transfer ability, but these models do not perform equally well in all languages. Evaluating task-specific performance of a model in a large number of languages is often a challenge due to lack of labeled data, as is targeting improvements in low performing languages through few-shot learning. We present a tool - LITMUS Predictor - that can make reliable performance projections for a fine-tuned task-specific model in a set of languages without test and training data, and help strategize data labeling efforts to optimize performance and fairness objectives.

Introduction

As developers build NLP systems for wider audiences, massively multilingual models such as mBERT (Devlin et al. 2019) and XLM-R (Conneau et al. 2020) are seeing greater adoption due to their cross-lingual zero-shot transfer ability (Turc et al. 2021; Choudhury and Deshpande 2021). However, developers face many practical challenges while deploying such systems. High-quality evaluation sets are typically available only in a few languages, and for a handful of tasks (Joshi et al. 2020). This is particularly troubling since zero-shot task performance varies significantly across languages (Hu et al. 2020; Wu and Dredze 2020; Pires, Schlinger, and Garrette 2019). Few-shot learning (Lauscher et al. 2020) and the choice of an appropriate pivot language (Turc et al. 2021; Liang et al. 2020; Lin et al. 2019) can significantly mitigate the poor zero-shot performance in a language. However, the performance gains per task-specific labeled example during fine-tuning stage varies widely across the tasks and languages. There are also notable positive and negative interferences between languages observed during the fine-tuning process. Thus, a second challenge for the developer is to decide how much labeled data should be collected and for which language(s), to achieve the desired performance in a set of languages.

We present an AI assistant - the LITMUS[1] Predictor - which helps a developer solve the above challenges by (1) making performance projections for a language (without labeled test data) given task-specific fine-tuned model built on a multilingual pre-trained model, and (2) suggesting how much data one should label in which languages such that desired performance objectives are met across a set of languages within a specified set of constraints. Thus, LITMUS Predictor helps a developer build *reliable* and high performing multilingual NLP models within a specified *budget*, other *constraints* and *fairness* objectives (Choudhury and Deshpande 2021). Further details on datasets, featurization, experiments and results can be found in Srinivasan et al. (2021). Our tool and code is released publicly to help users build large-scale multilingual models[2].

System Description

First, we will introduce the features of the LITMUS Predictor through its front-end, and then explain the back-end architecture.

FRONT-END. The tool takes as input (I1 in Fig. 1) the type of *pre-trained model* (currently we support mBERT or XLM-R, but it can be extended to any pre-trained model), a *task* (presently, XNLI, UDPOS or WikiANN, but can be customized for any task) and a set of *fine-tuning configurations*. A fine-tuning configuration C_k is specified as a set of pivot languages (henceforth pivots) and amount of labeled data in each of them: $\{\langle l_i, d_i \rangle\}_k$. In addition, the tool takes a set of *target languages* (henceforth, *targets*), $\{t_j\}$. The output of the tool are the projected performances, p_{kj}, of configuration C_k on target t_j, represented as a two dimensional heatmap (R2 in Fig. 1, green = better than, amber = equal and red = poorer than average). The performance of the best configuration (defined as $\text{argmax}_k \sum_j p_{kj}$) is highlighted (R1 in Fig. 1) along with the expected error in the predictions.

Data Labeling Plan. The user can also specify a *budget* b - the number of additional data points the user is willing to label, a set of languages, $\{l_j^A\}$, where further data can be

[*]These authors have equal contributions.
[†]Work done when the author was at Microsoft Research.

[1]LITMUS, an ongoing project at Microsoft, stands for Linguistically Informed Training & testing of MUltilingual Systems.
[2]Tool available at https://microsoft.github.io/Litmus, and code available at https://github.com/microsoft/Litmus.

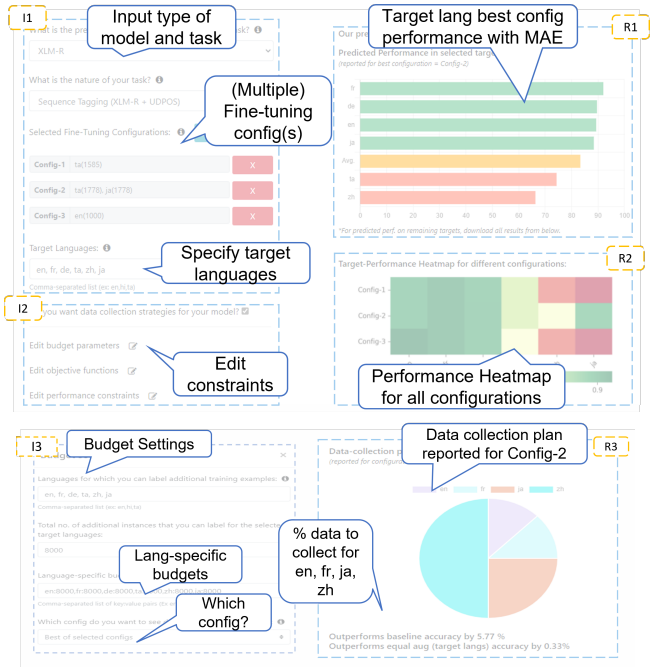

Figure 1: LITMUS interface (some panels omitted).

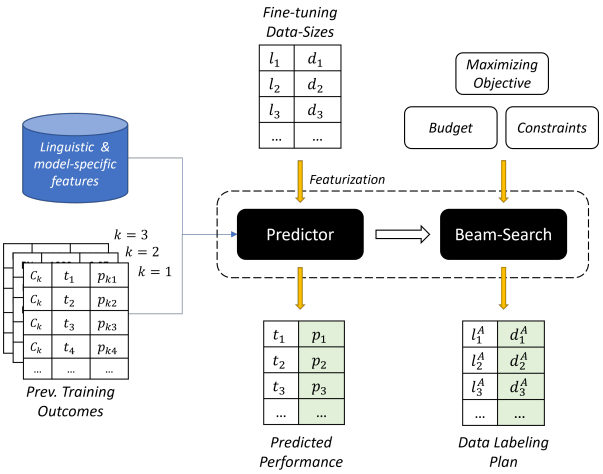

Figure 2: System Architecture

collected, language-specific budget caps, $\{b_j\}$, and the base configuration C^* for which a data labeling plan is required (I2 in Fig. 1). The data labeling plan can be customized for different maximizing objectives such as weighted average or minimum[3] performance across the targets, subjected to different constraints (such as target-specific or overall minimum or average performance). Based on these parameters, the tool suggests an optimal allocation, d_j^A, of the budget for language l_j^A (Fig. 1, presented as a pie-chart) along with the expected improvement in performance.

BACK-END. Fig. 2 shows the architecture of the system.

Performance Prediction: We learn the predictor by training an XGBoost[4]-based regression model on previous training outcomes, i.e., a configuration and target pair, $\langle C_k, t_j \rangle$, as input and the corresponding performance, p_{kj}, as the output. In addition to the fine-tuning data-sizes (d_j), the predictor model also uses features specific to the pre-trained model such as pre-training data size of the target, target-specific linguistic features obtained from WALS (Dryer and Haspelmath 2013), the syntactic distance between the target and pivot(s) computed as the cosine between the URIEL vectors (Littell et al. 2017), and the fraction of shared vocabulary in the pre-trained model between the target and pivot(s). Currently, the system has predictor models trained for three tasks – Natural language inferencing, POS tagging and Named entity recognition, using the XNLI (Conneau et al. 2018), UDPOS (Zeman et al. 2019) and WikiANN (Raiman and Raiman 2018) datasets, respectively.

Customization: There is option to build task and pre-trained LM specific custom predictors by providing appropriate training data as defined above.

Evaluation: We evaluate the predictor model under two settings, one where we assume that labeled test data is available for a language. In this case the mean average error of the predictions across targets (all languages in the dataset) are 0.61%, 0.85% and 0.89% for UDPOS, WikiANN and XNLI, respectively. In the second scenario, we assume that no labeled data is available for a target, and therefore, it does not appear as a pivot or target in any of the training instances. The average error across targets in this case are 4.62%, 8.08% and 9.93% for XNLI, UDPOS and WikiANN, respectively. Thus, in both cases the model's performance projections are fairly accurate.

Data Labelling Plan Generation: The trained predictor allows us to analyze the effect of adding fine-tuning data in different languages. By searching across the space of such possible data-augmentations subject to user's budget and other constraints, we can generate an optimal data labelling plan. Internally, we run a beam-search starting from the maximum allowable budget allocated to each l_j^A (If a user does not specify language specific budget, then the maximum allowable budget is simply b). We iteratively reduce the data allocation in different languages and retain the optimal configurations as per our trained predictor projections. We stop once the total augmentation across all languages is within the user's budget. Even though the prediction error on unseen targets is high for some tasks, the budget allocation is still effective because the system is able to predict the accuracy trends across targets correctly.

About the Demonstration. During the demonstration, apart from explaining the features and technology, we will also discuss some of the best practices and tips for training reliable and accurate performance predictors with limited training data.

[3]Maximizing the min across languages is a prioritarian or Rawlsian approach to fairness as opposed to average, which maximizes an utilitarian objective (Choudhury and Deshpande 2021).

[4]https://xgboost.ai/. Other ML algorithms are also being tried; so far, XGBoost has the best performance.

References

Choudhury, M.; and Deshpande, A. 2021. How Linguistically Fair are Multilingual Pre-trained Language Models? In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. Online: Association for Computational Linguistics.

Conneau, A.; Rinott, R.; Lample, G.; Williams, A.; Bowman, S.; Schwenk, H.; and Stoyanov, V. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2475–2485.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dryer, M. S.; and Haspelmath, M., eds. 2013. *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology.

Hu, J.; Ruder, S.; Siddhant, A.; Neubig, G.; Firat, O.; and Johnson, M. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Joshi, P.; Santy, S.; Budhiraja, A.; Bali, K.; and Choudhury, M. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the ACL*.

Lauscher, A.; Ravishankar, V.; Vulić, I.; and Glavaš, G. 2020. From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers. *arXiv preprint arXiv:2005.00633*.

Liang, Y.; Duan, N.; Gong, Y.; Wu, N.; Guo, F.; Qi, W.; et al. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*.

Lin, Y.-H.; Chen, C.-Y.; Lee, J.; Li, Z.; Zhang, Y.; Xia, M.; Rijhwani, S.; He, J.; Zhang, Z.; Ma, X.; et al. 2019. Choosing transfer languages for cross-lingual learning. *arXiv preprint arXiv:1905.12688*.

Littell, P.; Mortensen, D. R.; Lin, K.; Kairis, K.; Turner, C.; and Levin, L. 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*, 8–14.

Pires, T.; Schlinger, E.; and Garrette, D. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4996–5001.

Raiman, J.; and Raiman, O. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Srinivasan, A.; Sitaram, S.; Ganu, T.; Dandapat, S.; Bali, K.; and Choudhury, M. 2021. Predicting the Performance of Multilingual NLP Models. *arXiv preprint arXiv:2110.08875*.

Turc, I.; Lee, K.; Eisenstein, J.; Chang, M.-W.; and Toutanova, K. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Wu, S.; and Dredze, M. 2020. Are All Languages Created Equal in Multilingual BERT? *arXiv preprint arXiv:2005.09093*.

Zeman, D.; Nivre, J.; Abrams, M.; Aepli, N.; and al et. 2019. Universal Dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.