Multi Task Learning For Zero Shot Performance Prediction of Multilingual Models

Kabir Ahuja[1] * **Shanu Kumar**[2] * **Sandipan Dandapat**[2] **Monojit Choudhury**[1]
[1] Microsoft Research, India
[2] Microsoft R&D, Hyderabad, India
{t-kabirahuja,shankum,sadandap,monojitc}@microsoft.com

Abstract

Massively Multilingual Transformer based Language Models have been observed to be surprisingly effective on zero-shot transfer across languages, though the performance varies from language to language depending on the pivot language(s) used for fine-tuning. In this work, we build upon some of the existing techniques for predicting the zero-shot performance on a task, by modeling it as a multi-task learning problem. We jointly train predictive models for different tasks which helps us build more accurate predictors for tasks where we have test data in very few languages to measure the actual performance of the model. Our approach also lends us the ability to perform a much more robust feature selection, and identify a common set of features that influence zero-shot performance across a variety of tasks.

1 Introduction

Multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have been recently shown to be surprisingly effective for zero-shot transfer (Pires et al., 2019) (Wu and Dredze, 2019), where on fine-tuning for a task on one or a few languages, called *pivots*, they can perform well on languages unseen during training. The zero-shot performance however, is often not uniform across the languages and the multilingual models turn out to be much less effective for low resource languages (Wu and Dredze, 2020; Lauscher et al., 2020) and the languages that are typologically distant from the pivots (Lauscher et al., 2020). What affects the zero-shot transfer across different languages is a subject of considerable interest and importance (K et al., 2020; Pires et al., 2019; Wu and Dredze, 2019; Lauscher et al., 2020), however there is little conclusive evidence and a few papers even show contradictory findings.

Lauscher et al. (2020) recently, showed that it is possible to predict the zero shot performance of mBERT and XLM-R on different languages by formulating it as a regression problem, with pretraining data size and typological similarities between the pivot and target languages as the input features, and the performance on downstream task as the prediction target. Along similar lines Srinivasan et al. (2021) and Dolicki and Spanakis (2021) explore zero-shot performance prediction with a larger set of features and different regression techniques.

However, the efficacy of these solutions are severely limited by the lack of training data, that is, the number of languages for which performance metrics are available for a given task. For instance, for most tasks in the popular XTREME-R (Ruder et al., 2021) benchmark, there are data points for 7-11 languages. This not only makes zero-shot performance prediction a challenging problem, but also a very important one because for practical deployment of such multilingual models, one would ideally like to know its performance for all the languages the model is supposed to handle. As Srinivasan et al. (2021) shows, accurate performance predictors can also help us build better and fairer multilingual models by suggesting data labeling strategies.

In this work, we propose multi-task learning (Zhang and Yang, 2017) as an approach to mitigate training-data constraints and consequent over-fitting of the performance predictors to tasks and/or datasets. The contributions of our work are fourfold. *First,* we experiment with different multi-task learning approaches, such as Group Lasso (Yuan and Lin, 2006), Collective Matrix Factorization (Cortes, 2018), Multi-Task Deep Gaussian Process Regression (Bonilla et al., 2008) and Meta Agnostic Meta Learning (Finn et al., 2017) for 11 tasks. We observe an overall 10% reduction in performance prediction errors compared to the best performing single-task models. The gains are even stronger when we just consider the tasks with very few data points (≤ 10), where we see a 20%

[*] Equal contribution

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5454 - 5467
May 22-27, 2022 ©2022 Association for Computational Linguistics

drop in the mean absolute errors. *Second,* an interesting consequence of modelling this problem via multi-task learning is that we are able to predict performance on low resource languages much more accurately, where in some cases single-task approaches may perform even worse than the simple averaging baselines. *Third,* apart from the features used for zero-shot performance prediction in the previous work (Lauscher et al., 2020; Srinivasan et al., 2021; Dolicki and Spanakis, 2021), we also utilize metrics quantifying the quality of multilingual tokenizers as proposed in (Rust et al., 2021) as features in our predictive models, which turn out to have strong predictive power for certain tasks. To the best of our knowledge, our work is the first to explore the impact of tokenizer quality specifically on zero-shot transfer. And *fourth,* our multi-task framework in general lends us with a much more robust selection of features affecting the zero-shot performance. This, in turn, lets us investigate the critical open question on what influences the zero-shot performances across languages more rigorously. As we shall see, our findings corroborate some of the previous conclusions, while others are extended or annulled.

2 Background and Related Work

Zero Shot Transfer. Multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have shown surprising effectiveness in zero-shot transfer, where fine-tuning the MMLM on a task in some source language often leads to impressive performance on the same task in other languages as well without explicitly training on them. Pires et al. (2019) first observed this phenomenon for NER (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003; Levow, 2006) and POS tagging (Nivre et al., 2018) tasks. Concurrently, Wu and Dredze (2019) also showed this surprisingly cross lingual transfer ability of mBERT additionally on tasks like Document Classification (Schwenk and Li, 2018), Natural Language Inference (Conneau et al., 2018) and Dependency Parsing (Nivre et al., 2018).

Factors Affecting Zero Shot Transfer. Pires et al. (2019) showed that vocabulary memorization played little role in zero-shot generalization as language pairs with little word piece overlap also exhibited impressive crosslingual performance. K et al. arrived at a similar conclusion by training BERT on an artificially generated language to zero out the word overlap with the target languages, and observed only minor drops in the performance compared to training the model on English. On the contrary Wu and Dredze (2019), observed strong correlations between the sub-word overlap and the zero-shot performance in four out of five tasks.

Wu and Dredze (2020) showed that mBERT performed much worse for zero-shot transfer to low resource languages (i.e., less pre-training data) than high resource ones on POS Tagging, NER and Dependency Parsing tasks. Lauscher et al. (2020) also had a similar observation on tasks like XNLI and XQuAD (Artetxe et al., 2020), though they found that the zero-shot performance on NER, POS tagging and Dependency Parsing tasks might not strictly depend on the pre-training size and could be better explained by different linguistic relatedness features like syntactic and phonological similarities between the language pair. Similar dependence on the typological relatedness such as word order had also been observed by Pires et al. (2019).

Performance Prediction. Prior work has explored predicting the performance of machine learning models from unlabelled data by either measuring (dis)agreements between multiple classifiers (Platanios et al., 2014, 2017) or by utilizing underlying information about data distribution (Domhan et al., 2015). In the context of NLP Birch et al. (2008) explored predicting the performance of a Machine Translation system by utilizing different explanatory variables for the language pairs. Lin et al. (2019) proposed a learning to rank approach to choose transfer languages for cross lingual learning using several linguistic and dataset specific features.

Recently, there has been an interest in predicting the performance of NLP models without actually training or testing them, by formulating it as a regression problem. Xia et al. (2020) showed that using experimental settings for an NLP experiment as inputs it is possible to accurately predict the performance on different languages and model architectures.Ye et al. (2021) extended this work by proposing methods to do a fine-grained estimation of the performance as well as predicting well-callibrated confidence intervals. Specifically predicting the zero-shot performance of MMLMs was first explored in Lauscher et al. (2020), where they used a linear regression model to estimate the cross-lingual transfer performance based on pre-training data size and linguistic relatedness features.

Srinivasan et al. (2021) tackled this problem by utilizing XGBoost Regressor for the prediction along with a larger set of features. Dolicki and Spanakis (2021) explored individual syntactic features for zero-shot performance prediction instead of working with aggregate similarity values, and showed about 2 to 4 times gain in performance. We extend all of these works by considering a multi-task learning approach, where performance prediction in a task utilizes not only the data available for that task, but also the patterns observed for other tasks.

3 Problem Setup

We begin by defining the multi-task performance prediction problem and then describe the different linguistic and MMLM specific features used.

3.1 Multi-Task Performance Prediction Problem

Consider a pre-trained multilingual model $\mathcal{M}$, trained using self supervision on a set of languages $\mathcal{L}$. Let $\mathfrak{T}$ be the set of downstream NLP tasks, $\mathcal{P}$ be the set of pivot (source) languages for which training data is available for the downstream tasks for fine-tuning and $\mathcal{T}$ be the set of target languages for which validation/test data is available. Note that $\mathcal{P} \subset \mathcal{L}$ and $\mathcal{T} \subseteq \mathcal{L}$. We use the zero-shot setting similar to Lauscher et al. (2020) which enforces $\mathcal{P}$ and $\mathcal{T}$ to be disjoint sets[1], i.e., $\mathcal{P} \cap \mathcal{T} = \emptyset$.

We then define $y_{p,t}^{\mathcal{M},\mathfrak{t}} \in \mathbb{R}$ as the zero-shot performance on language $t \in \mathcal{T}$ on finetuning $\mathcal{M}$ on task $\mathfrak{t} \in \mathfrak{T}$ in pivot language $p \in \mathcal{P}$. Let $x_{p,t}^{\mathcal{M}} \in \mathbb{R}^n$ be the n-dimensional feature vector representing the corresponding train-test configuration. Since for our experiments we train and evaluate the performance prediction for a single model at a time, we will simplify the notations to $y_{p,t}^{\mathfrak{t}}$ and $x_{p,t}$.

The predictor model can then be defined as the function $f_{\Theta,\Phi} : \mathbb{R}^n \times \mathfrak{T} \rightarrow \mathbb{R}$, where $\Theta \in \mathbb{R}^{d_g}$ denotes the shared parameters across the tasks and the task specific parameters are given by $\Phi \in \mathbb{R}^{d_s \times |\mathfrak{T}|}$. The objective function for training such a predictor model can be defined as:

$$J(\Theta, \Phi) = \sum_{\mathfrak{t} \in \mathfrak{T}} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \|f(x_{p,t}, \mathfrak{t}; \Theta, \Phi) - y_{p,t}^{\mathfrak{t}}\|_2^2 + \lambda_g \|\Theta\|_1 + \lambda_s \|\Phi\|_{1,1} + \lambda_{group} \|\Phi\|_{1,q} \tag{1}$$

The second and third terms regularize the global and task specific parameters independently, while the last term, l_1/l_q norm with $q > 1$, ensures a block sparse selection of the task specific parameters. This term ensures a multi-task learning behavior even when there are no parameters shared across the tasks (i.e., $\Theta = \emptyset$) through selection of common features across the tasks. Setting $\Theta = \emptyset$ and $\lambda_{group} = 0$ leads to the single task setup of Lauscher et al. (2020) and Srinivasan et al. (2021).

3.2 Features

We divide the set of features into two higher level categories, viz. the pairwise features defined for the pivot and target that measure the typological relatedness of the languages, and the individual features defined for the target language reflecting the state of its representation in $\mathcal{M}$.

3.2.1 Pairwise Features

Instead of directly using the different typological properties of the the two languages as features, we use the pairwise relatedness to avoid feature explosion.

Subword Overlap : We define the subword overlap as the percentage of unique tokens that are common to the vocabularies of both the pivot and target languages. Let V_p and V_t be the subword vocabularies of p and t. The subword overlap is then defined as :

$$o_{sw}(p, t) = \frac{|V_p \cap V_t|}{|V_p \cup V_t|} \tag{2}$$

Similarity between Lang2Vec vectors: Following Lin et al. (2019) and Lauscher et al. (2020), we compute the typological relatedness between p and t from the linguistic features provided by the URIEL project (Littell et al., 2017). We use syntactic ($s_{syn}(p, t)$), phonological similarity ($s_{pho}(p, t)$), genetic similarity ($s_{gen}(p, t)$) and geographic distance ($d_{geo}(p, t)$). For details, please see Littell et al. (2017).

3.2.2 Individual Features

Pre-training Size: We use the log_{10} of the size (in words) of the pre-training corpus in the target language, SIZE(t), as a feature.

Rare Typological Traits: Srinivasan et al. (2021) proposed this metric to capture the rarity of the typological features of a language in the representation of $\mathcal{M}$. Every typological feature in WALS

[1] Though beyond the scope of the current work, it is possible to extend this to a few-shot setting as discussed in Srinivasan et al. (2021).

database is ranked based on the amount of pretraining data for the languages that contain the feature. For the language t, Mean Reciprocal Rank (MRR) of all of its features is then calculated and used as a feature – WMRR(t).

Tokenizer Features : In their recent work, Rust et al. (2021) proposed two metrics, viz. tokenizer's *fertility* and proportion of continued words, to evaluate the quality of multilingual tokenizers on a given language. For target t, they define the tokenizer's fertility, FERT(t), as the average number of sub-words produced for every tokenized word in t's corpus. On the other hand, the proportion of continued words, PCW(t), measures how often the tokenizer chooses to continue a word across at least two tokens. They show that the multilingual models perform much worse on a task than their monolingual counterparts when the values of these metrics are higher for the multilingual tokenizer. We include FERT(t) and PCW(t) as features.

An important thing to note here is that the we do not use identity of a language as a feature while training the models, hence the performance prediction models are capable of generating predictions on new languages unseen during training. However, if the features of the new languages deviate significantly from the features seen during training, the predictions are expected to be less accurate as also observed in Xia et al. (2020); Srinivasan et al. (2021) and is one of the main reasons for exploring a multi-task approach.

4 Approaches

We extensively experiment with a wide-array of multi-task as well as single-task regression models to provide a fair comparison between different approaches to zero-shot performance prediction.

4.1 Baselines

Average Score Within a Task (AWT) : The performance for a pivot-target pair (p , t) on a task $\mathfrak{t}$ is approximated by taking the average of the performance on all other target languages (pivot being fixed) in the same task $\mathfrak{t}$, i.e., $f(x_{p,t}, \mathfrak{t}) = \frac{1}{|\mathcal{T}|-1} \sum_{t' \in \mathcal{T}-\{t\}} y^{\mathfrak{t}}_{p,t'}$.

Average Score across the Tasks (AAT) : Here instead of averaging over all the target languages within a task, we approximate the performance on a given target language by averaging the scores for that language across the other tasks, i.e., $f(x_{p,t}, \mathfrak{t}) = \frac{1}{|\mathfrak{T}|-1} \sum_{\mathfrak{t}' \in \mathfrak{T}-\{\mathfrak{t}\}} y^{\mathfrak{t}'}_{p,t}$.

4.2 Single Task Models

Lasso Regression: Lauscher et al. (2020) train different linear regression models for each task. Along similar lines, we experiment with linear regression, but also add an L1 regularization term, as we observed it usually leads to better predictors.

XGBoost Regressor: As shown in Srinivasan et al. (2021), XGBoost (Chen and Guestrin, 2016) generally obtains impressive performance on this task, and hence we include it in our experiments as well.

4.3 Multi Task Models

Group Lasso: l_1/l_q norm based block-regularization has been shown to be effective for multi-task learning in the setting of multi-linear regression (Yuan and Lin, 2006; Argyriou et al., 2008). For each task, consider separate linear regression models represented by the weight matrix $\Phi \in \mathbb{R}^{n \times |\mathfrak{T}|}$. The l_1/l_q regularization term is given as: $\|\Phi\|_{1,q} = \sum_{j=1}^{n} (\sum_{\mathfrak{t}=1}^{|\mathfrak{T}|} |\Phi_{j\mathfrak{t}}|^q)^{1/q}$, where $\Phi_{j\mathfrak{t}}$ denotes the weight for the feature j in the task $\mathfrak{t}$. For $q > 1$, minimizing this term pushes the l_q-norms corresponding to the weights of a given feature across the tasks to be sparse, which encourages multiple predictors to share similar sparsity patterns. In other words, a common set of features is selected for all the tasks. We use $q = 2$ for the group regularization term.

Since this can be restrictive in certain scenarios, some natural extensions to Group Lasso, such as Dirty Models (Jalali et al., 2010) and Multi Level Lasso (Lozano and Swirszcz, 2012), have been proposed that separate out the task specific and global parameters. We experimented with these methods and observed equivalent or worse performance compared to Group Lasso.

Collective Matrix Factorization (CMF) with Side Information: Low rank approximation for the task weights matrices forms one family of methods for multi-task learning (Zhang and Yang, 2017; Pong et al., 2010; Ando et al., 2005). As a direct analogue with collaborative filtering, here we can think of the tasks as *users* and pivot-target pairs as *items*. Consider the matrix $\mathbf{Y} \in \mathbb{R}^{|\mathfrak{T}| \times |\mathcal{P} \times \mathcal{T}|}$, where each element of the matrix correspond to $y^{\mathfrak{t}}_{p,t}$. We can then decompose the matrix into task and language-pair specific factors as

$$\mathbf{Y} \sim \mathbf{T}\mathbf{L}^T \tag{3}$$

where $\mathbf{T} \in \mathbb{R}^{|\mathfrak{T}| \times d_{latent}}$ and $\mathbf{L} \in \mathbb{R}^{|\mathcal{P} \times \mathcal{T}| \times d_{latent}}$ are the task and language-pair factor matrices, and d_{latent} is the number of factors.

Additionally, in order to incorporate the feature information about the language pairs as discussed in section 3.2, we incorporate Collective Matrix Factorization approach (Cortes, 2018). It incorporates the attribute information about items and/or users in the factorization algorithm by decomposing the language-pair feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{P} \times \mathcal{T}| \times n}$ as $\mathbf{LF}^T$, such that $\mathbf{L}$ is shared across both decompositions. This helps to learn the latent representations for the pivot-language pairs from the task-wise performance as well as different linguistic and MMLM specific features[2]. In relation to Equation 1, we can think of task factors $\mathbf{T}$ to correspond to the task specific parameters Φ, language-pair factors $\mathbf{L}$ as the shared parameters Θ and the predictor model as $f(x_{p,t}, \text{t}; \Theta, \Phi) = (\mathbf{TL}^T)_{(p,t),\text{t}}$. Both $\mathbf{L}$ and $\mathbf{T}$ are regularized seperately, but there is no group regularization term ($\lambda_{group} = 0$).

Ye et al. (2021) also uses a Tensor Factorization approach for performance prediction which is similar to our CMF method. However, they train separate models for each task and factorize over metric specific attributes instead for a fine-grained prediction.

Multi-Task Deep Gaussian Process Regression (MDGPR): We use the multi-task variant of Gaussian Processes proposed in Bonilla et al. (2008) and utilize deep neural networks to define the kernel functions as in Deep GPs (Wilson et al., 2016). For comparison, we also report the scores of the single-task variant of this method which we denote as DGPR. See Appendix (section A.1) for details.

Apart from these we also explore other multi-task methods like Model Agnostic Meta Learning (MAML) (Finn et al., 2017), details of which we leave in the appendix (section A.1).

5 Experimental Setup

In this section, we discuss our test conditions, datasets and training parameters for the different experiments.

5.1 Test Conditions

We consider two different test conditions: Leave One Language Out (LOLO) and Leave Low Resource Languages Out (LLRO).

Leave One Language Out: LOLO is a popular setup for multilingual performance prediction (Lauscher et al., 2020; Srinivasan et al., 2021), where for a given task, we choose a target language and move all of its instances from the prediction dataset to the test data. The models are then trained on the remaining languages and evaluated on the unseen test language. This is done for all the target languages available for a task, and the Mean Absolute Error (MAE) across languages is reported. In the multi-task setting we evaluate on one task at a time while considering the rest as helper tasks for which the entire data is used including the test language[3].

Leave Low Resource Languages Out: Through this evaluation strategy we try to emulate the real world use case where we only have test data available in high resource languages such as English, German and Chinese, and would like to estimate the performance on under-represented languages such as Swahili and Bengali. We use the language taxonomy provided by Joshi et al. (2020) to categorize the languages into six classes (0 = low to 5 = high) based on the number of resources available. We then move languages belonging to class 3 or below to our test set and train the models on class 4 and 5 languages only. Similar to LOLO, here too we allow the helper tasks to retain all the languages.

5.2 Tasks and Datasets

We use the following 11 tasks provided in XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021) benchmarks: 1. **Classification**: XNLI (Conneau et al., 2018) , PAWS-X (Yang et al., 2019), and XCOPA (Ponti et al., 2020) 2. **Structure Prediction**: UDPOS (Nivre et al., 2018), and NER (Pan et al., 2017) 3. **Question Answering**: XQUAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), and TyDiQA-GoldP (Clark et al., 2020) 4. **Retrieval**: Tatoeba (Artetxe and Schwenk, 2019), Mewsli-X (Botha et al., 2020; Ruder et al., 2021), and LAReQA (Roy et al., 2020)

All of these datasets have training data present only in English i.e. $\mathcal{P} = \{\texttt{en}\}$, and majority of the tasks have fewer than 10 target languages.

[2] Note that we can use a similar approach for providing side information for the tasks as well.

[3] Note that this is a reasonable relaxation to make as it is closer to the real world use case where we would have the evaluation data for some languages in the standard tasks and would like to utilize that to make predictions on the same languages for the new ftask.

MMLM	Task	$\|\mathcal{T}\|$	Baselines		Single Task Models			Multi Task Models			
			Average within Task	Average across Tasks	Lasso	XGBoost	DGPR	Group Lasso	CMF	MDGPR	MAML
XLMR	**MLQA**	7	2.92	2.26	4.33	2.91	3.26	**2.21**	2.66	2.96	4.89
	PAWS	7	3.34	0.9	**0.8**	1.28	1.27	1.32	1.39	2.71	6.62
	XCOPA	8	4.52	5.91	2.42	4.16	4.73	2.69	2.03	**1.96**	6.28
	TyDiQA	9	4.29	5.48	5.89	5.63	6.56	5.04	5.88	**4.61**	4.96
	XQUAD	10	4.90	4.22	4.54	6.56	4.13	4.16	3.86	**3.15**	6.85
	LAReQA	10	2.10	**1.51**	1.53	1.56	1.78	1.52	1.87	2.69	8.22
	MewsliX	10	16.61	15.48	15.70	21.16	15.66	13.73	14.62	10.07	**9.33**
	XNLI	14	3.07	2.07	1.97	**1.53**	2.16	2.17	2.17	3.54	4.55
	WikiANN	32	15.22	11.61	10.14	10.26	12.64	10.92	11.36	**9.15**	13.19
	Tatoeba	35	8.69	8.68	**5.82**	7.14	6.80	5.83	6.08	8.09	9.72
	UDPOS	48	10.15	7.65	7.52	**5.12**	6.02	7.72	7.89	5.88	10.71
	Average	19	6.89	5.98	5.51	6.12	5.91	5.21	5.44	**4.98**	7.76
	Average ($\|\mathcal{T}\| \leq 10$)	9	5.53	5.11	5.03	6.18	5.34	4.38	4.62	**4.02**	6.73
mBERT	**Average**	19	8.69	6.57	5.55	6.86	6.10	5.45	**5.08**	5.12	8.14
	Average ($\|\mathcal{T}\| \leq 10$)	9	6.96	5.64	4.99	6.54	5.73	4.44	**4.18**	4.53	7.51

Table 1: Mean Absolute Error (scaled by 100 for readability) for LOLO for different approaches across tasks. We also report the average MAE across all tasks ("Average") and for tasks which has less than or equal to 10 languages ("Average ($|\mathcal{T}| \leq 10$)"). Task-wise results for mBERT can be found in the Appendix (table 2)

5.3 Training Details

We train and evaluate our performance prediction models for **mBERT** (*bert-base-multilingual-cased*) and **XLM-R** (*xlm-roberta-large*). For training XGBoost, we used 100 estimators with a maximum depth of 10. For Group Lasso, we used the implementation provided in the MuTaR software package[4], and used a regularization strength of 0.01. We optimized CMF's objective function using Alternating Least Squares (ALS), used 5 latent factors with a regularization parameter equal to 0.1, and used the Collective Matrix Factorization python library[5]. In case of MDGPR, we used Radial Basis Function as the kernel and a two-layer MLP for learning latent features, with 50 and 10 units followed by ReLU activation. We set the learning rate and epochs as 0.01 and 200, and implemented it using GPyTorch[6].

6 Results and Discussion

6.1 LOLO Results

Table 1 shows MAE (in %) for LOLO for different single-task and multi-task models on the tasks. For XLMR, we observe that multi-task models, primarily MDGPR, often outperform the best single-task models by significant margins, and for tasks like MewsliX we even see about 36% reduction in MAE. Overall, we see about 10% drop in LOLO errors on average for MDGPR compared to the best performing single-task model i.e. Lasso Regression. As expected, the benefit of multi-task learning is even more prominent when we consider the tasks for which only a few (≤ 10) data points are available. Here we see about 20% reduction in errors. For mBERT as well, we have similar observations, except that CMF performs slightly better than MDGPR.

Note that the *Average across task* baseline is quite competitive and performs better than single-task XGBoost and MAML in average, and better than all models for LAReQA.

Figure 2 plots the dependence of the number of helper tasks on the performance of the multi-task models. As expected, MAE decreases as helper tasks increase, especially for MDGPR and CMF. On a related note, the Pearson Correlation coefficient between MAE and number of tasks a target language is part of is found to be -0.39, though the trend in this case is not as clear.

6.2 LLRO Results

Predicting the performance on low resource languages, for which often standard training and test datasets are not available, can be an important use case where multi-task performance prediction can be helpful. Figure 6 in appendix shows the class-wise (Joshi et al., 2020) distribution of languages for the tasks that we consider in our experiments. As one would expect, for most tasks, test data is available for languages belonging to class-4 and class-5. Training performance prediction models without any task to transfer from can therefore, possibly lead to poor generalization on the low resource languages. On the other hand, for the same reason - lack of test data, building accurate predictors for low-resource languages is necessary.

[4] https://github.com/hichamjanati/mutar

[5] https://github.com/david-cortes/cmfrec

[6] https://gpytorch.ai/

MAE values for the LLRO evaluation setup are shown in figure 1 for XLMR. Results for mBERT follow similar trends and are reported in the Appendix (figure 7). For both XLMR and mBERT we observe that the three main multi-task models – Group Lasso, CMF and MDGPR – outperform the single-task models and baselines. Interestingly, for XLMR, the single task models XGBoost and Lasso perform even worse than the *Average within Tasks* baseline. Overall we see around 18% and 11% drop in MAE for Group Lasso over the best performing single-task model, for XLMR and mBERT respectively.

6.3 Feature Importance

An interesting consequence of zero-shot performance prediction is that the models can be directly used to infer the correlation (and possibly causation) between linguistic relatedness and pretraining conditions and zero-shot transferability. Multi-task learning, in this context, help us make more robust inferences, as the models are less prone to overfitting to a particular task or dataset.

Figure 3 shows the SHAP values of the features for the Group Lasso model trained on XLMR's zero-shot performance data. As expected for Group Lasso, we see a block-sparsity behavior among the tasks. Features such as Rare Typological Traits (WMRR(t)), Tokenizer's Fertility (FERT(t)) and Genetic Similarity ($s_{gen}(p,t)$) are ignored in all the tasks. In contrast, for the single-task lasso regression (Figure 9 in Appendix), we see different sets of features selected for different tasks, which for the scale at which we operate, might not be indicative of the actual factors that affect the zero-shot performance in these tasks.

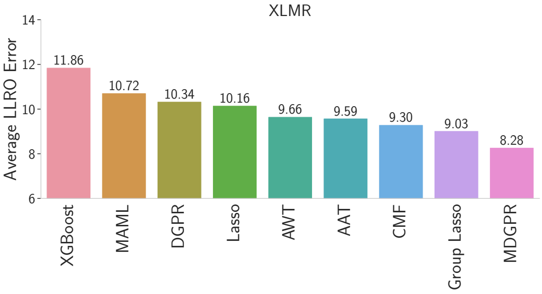


Figure 1: Leave Low Resource Out (LLRO) results for XLMR

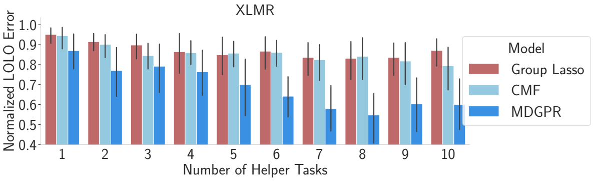
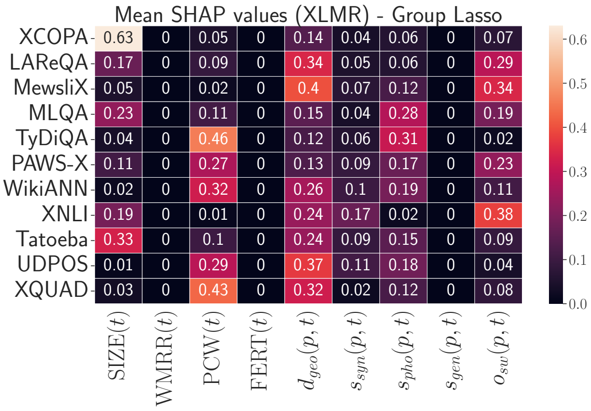


Figure 2: Number of helper tasks vs. LOLO MAE. Errors for different model types (Group Lasso, CMF and MDGPR) and tasks are scaled by diving them by the maximum error value.



	SIZE(t)	WMRR(t)	PCW(t)	FERT(t)	$d_{geo}(p,t)$	$s_{syn}(p,t)$	$s_{pho}(p,t)$	$s_{gen}(p,t)$	$o_{sw}(p,t)$
XCOPA	0.63	0	0.05	0	0.14	0.04	0.06	0	0.07
LAReQA	0.17	0	0.09	0	0.34	0.05	0.06	0	0.29
MewsliX	0.05	0	0.02	0	0.4	0.07	0.12	0	0.34
MLQA	0.23	0	0.11	0	0.15	0.04	0.28	0	0.19
TyDiQA	0.04	0	0.46	0	0.12	0.06	0.31	0	0.02
PAWS-X	0.11	0	0.27	0	0.13	0.09	0.17	0	0.23
WikiANN	0.02	0	0.32	0	0.26	0.1	0.19	0	0.11
XNLI	0.19	0	0.01	0	0.24	0.17	0.02	0	0.38
Tatoeba	0.33	0	0.1	0	0.24	0.09	0.15	0	0.09
UDPOS	0.01	0	0.29	0	0.37	0.11	0.18	0	0.04
XQUAD	0.03	0	0.43	0	0.32	0.02	0.12	0	0.08

Figure 3: Task-wise mean SHAP values of different features for the Group Lasso model trained on XLMR zero-shot performance data. Higher value implies stronger effect.

Subword Overlap. Among the features that get selected for all tasks, we observe that Subword Overlap ($o_{sw}(p,t)$) typically gets higher importance in retrieval (LAReQA and MewsliX) and sentence classification tasks (PAWS-X, XNLI). Since the retrieval tasks that we consider, as described in Ruder et al. (2021), measure the alignment between the cross lingual representations of semantically similar sentences, having a shared vocabulary between the languages can leak information from one to another (Wu and Dredze, 2019) which might improve the retrieval performance. Interestingly, if we compare this with the feature importance scores for the single task lasso model (Figure 9 in Appendix), we do see MewsliX task getting higher importance for the subword overlap, but LAReQA gets virtually zero SHAP value for this feature, showcasing how single-task models can misinterpret two similar tasks as requiring very different features. Our observation reinforce the generally held notion that vocabulary overlap between the pivot and target is beneficial for zero-shot transfer (Wu and Dredze, 2019), especially for retrieval tasks, though some studies have argued otherwise (Pires et al., 2019; K et al., 2020).

Tokenizer Features. For structure prediction (UDPOS and WikiAnn) and question answering

(XQUAD and TyDiQA) tasks that require making predictions for each token in the input, we see that the tokenizer feature, PCW(t), receive a higher SHAP value. In contrast, for single-task lasso, here too we do not observe high importance of this feature across these related tasks. Rust et al. (2021) note that languages such as Arabic where mBERT's multilingual tokenizer was found to be much worse than it's monolingual counterpart, there was a sharper drop in performance of mBERT compared to the monolingual model for QA, UDPOS and NER tasks than for sentiment classification. We believe that XLMR's surprisingly worse performance than mBERT for Chinese and Japanese on UDPOS might be correlated with it's significantly worse tokenizer for these languages based on the *fertility* (FERT) and *Percentage Continued Words* (PCW) feature values (see Appendix A.2 for exact values). The high SHAP values for PCW(t) further strengthen our belief[7].

Pre-training Size. Similar to the findings of Lauscher et al. (2020), we observe that pre-training corpus size has low SHAP value, and therefore, lower importance for lower level tasks such as UDPOS and NER, and higher SHAP values for higher level tasks like XNLI. Additionally, we extend their observations to tasks such as XCOPA, Tatoeba, MLQA and LAReQA where pre-training size seem to play a significant role in the performance prediction. Again, compared to single Lasso Regression model, we see a different selection pattern: Pre-training size receives a high SHAP value for UDPOS while for XNLI it is negligible. This neither fully conforms with our observations on the multi-task feature selections, nor with the previous work (Lauscher et al., 2020).

Typological Relatedness Features. Out of all the typological relatedness features, we found Geographical Distance ($d_{geo}(p,t)$) receiving highest SHAP values for all tasks, implying that geographical proximity between the pivot-target pair is an important factor in determining the zero-shot transferability between them. Lauscher et al. (2020) also observe positive correlations between geographical relatedness and zero-shot performance. The cross-task importance of geographic distance (unlike the other relatedness features) might be attributed to the 100% coverage across languages for the geographical vectors in the URIEL database. In contrast, Syntactic and Phonological vectors have missing values for a majority of the languages (Littell et al., 2017).

Like Lauscher et al. (2020), we also see some dependence on syntactic ($s_{syn}(p,t)$) and phonological ($s_{pho}(p,t)$) similarities for XLMR's zero shot performance on XNLI and XQUAD tasks respectively. However, in both cases we found that the tokenizer feature PCW(t) receives a much higher SHAP value. Interestingly, genetic similarity ($s_{gen}(p,t)$) is not selected for any task, arguably due to the block sparsity in feature selection of Group Lasso. We do see some tasks receiving high SHAP values for $s_{gen}(p,t)$ in single-task lasso (Figure 9 in Appendix). However, the number of such tasks as well as the SHAP values are on the lower side, implying that genetic similarity might not provide any additional information for zero-shot transfer over and above the geographical, syntactic and phonological similarities.

Similar trends are observed in the case of mBERT as well (Figure 10 in appendix), with some minor differences. For instance, instead of PCW(t), FERT(t) receives higher SHAP value; $s_{syn}(p,t)$ also receives higher importance, especially for tasks like UDPOS and XNLI, which is consistent with the findings of Lauscher et al. (2020).

7 Conclusion and Future Work

In this paper, we showed that the zero-shot performance prediction problem can be much more effectively and robustly solved by using multi-task learning approaches. We see significant reduction in errors compared to the baselines and single-task models, specifically for the tasks which have test sets available in a very few languages or when trying to predict the performance for low resource languages. Additionally, this approach allows us to robustly identify factors that influence zero-shot performance. Our findings in this context can be summarized as follows.

1. *Subword overlap* between the pivot and target has a strong positive influence on zero-shot transfer, especially for Retrieval tasks. 2. *Quality of the target tokenizer*, defined in terms of how often or how aggressively it splits the target tokens negatively influences zero-shot performance for word-level tasks such as POS tagging and Span extraction. 3. *Pre-training size* of the target positively

[7] Note that Rust et al. (2021) shows the importance of tokenizer metrics for the case where the multilingual models are fine-tuned on the target language, whereas we analyze their importance for zero-shot transfer.

influences zero-shot performance in many tasks, including XCOPA, Tatoeba, MLQA and LAReQA. 4. *Geographical proximity* between pivot and target is found to be uniformly important across all the tasks, unlike syntactic and phonological similarities, which are important for only some tasks.

This last finding is especially interesting. As described earlier, geographical proximity is a more clear, noise-free and complete feature compared to the other relatedness metrics. However, one could also argue that since neighboring languages tend to have high vocabulary and typological feature overlap due to contact processes and shared areal features, geographical distance is an extremely informative feature for zero-shot transfer. Two direct implications of these findings are: (1) for effective use of MMLMs, one should develop resources in at least one pivot language per geographic regions, and (2) one should work towards multilingual tokenizers that are effective for most languages.

There are a number of directions that can be explored in future related to our work. The prediction models can be extended to a multi-pivot and few-shot settings, as described in Srinivasan et al. (2021). Further probing experiments could be designed to understand the role of sub-word overlap on zero-shot transfer of Retrieval tasks.

Acknowledgements

We would like to thank the LITMUS team at Microsoft for their valuable inputs and feedback over the course of this project.

References

Rie Kubota Ando, Tong Zhang, and Peter Bartlett. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11).

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine learning*, 73(3):243–272.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of ACL 2020*.

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the ACL 2019*.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Honolulu, Hawaii. Association for Computational Linguistics.

Edwin V Bonilla, Kian Chai, and Christopher Williams. 2008. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, 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. In *Transactions of the Association of Computational Linguistics*.

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 EMNLP 2018*, pages 2475–2485.

David Cortes. 2018. Cold-start recommendations in collective matrix factorization. *arXiv preprint arXiv:1809.00366*.

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.

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*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

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 generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep Ravikumar. 2010. A dirty model for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

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 *International Conference on Learning Representations*.

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.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of ACL 2020*.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. 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.

Aurelie C Lozano and Grzegorz Swirszcz. 2012. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 595–602.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. 2018. Universal dependencies 2.2.

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 ACL 2017*, pages 1946–1958.

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.

Emmanouil Platanios, Hoifung Poon, Tom M Mitchell, and Eric J Horvitz. 2017. Estimating accuracy from unlabeled data: A probabilistic logic approach. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Emmanouil Antonios Platanios, Avrim Blum, and Tom Mitchell. 2014. Estimating accuracy from unlabeled data.

Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. 2010. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489.

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.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. 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.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Anirudh Srinivasan, Sunayana Sitaram, Tanuja Ganu, Sandipan Dandapat, Kalika Bali, and Monojit Choudhury. 2021. Predicting the performance of multilingual nlp models. *arXiv preprint arXiv:2110.08875*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. 2016. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain. PMLR.

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.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of EMNLP 2019*, pages 3685–3690.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, PP.

A Appendix

A.1 Additional Details of Approaches Used

Gaussian Process Regression (GPR): We start by briefly reviewing Gaussian Processes (GP) in context of the zero-shot performance prediction problem. For a pivot-target language pair (p, t) and a task $\mathfrak{t}$, the GP prior and the likelihood function can be defined as:

$$f \sim \mathcal{N}(\mu^{\mathfrak{t}}, K^{\mathfrak{t}}); \quad y|f(x_{p,t}) \sim \mathcal{N}(y^{\mathfrak{t}}_{p,t}; f(x_{p,t}), \sigma^2_{\mathfrak{t}}) \tag{4}$$

where $\mu^{\mathfrak{t}}$ is the mean and $K^{\mathfrak{t}}_{(p,t),(p',t')} = k^{\mathfrak{t}}(x_{p,t}, x_{p',t'})$ is the kernel of the GP defined on the task $\mathfrak{t}$. $\sigma^2_{\mathfrak{t}}$ denotes the noise variance.

Deep Gaussian Process Regression (DGPR): We use DGP (Wilson et al., 2016) to learn rich features from the observed data. Specifically, the kernel $k^{\mathfrak{t}}(x_{p,t}, x_{p',t'})$ now takes the transformed inputs as

$$k^{\mathfrak{t}}(x_{p,t}, x_{p',t'}) = k^{\mathfrak{t}}(g(x_{p,t}), g(x_{p',t'})) \tag{5}$$

where $g(x)$ is a non-linear mapping given by a deep network. Please refer to Wilson et al. (2016) for a detail account on optimization of DGP.

Multi-Task Deep Gaussian Process Regression (MDGPR): We use the multi-task variant of Gaussian Processes proposed in Bonilla et al. (2008)

where inter-task similarities are learnt solely based on the task identities and the observed data for each task. Instead of learning task-specific kernels $k^{\mathfrak{t}}(g(x_{p,t}), g(x_{p',t'}))$, we will have a common kernel over the inputs as $k(g(x_{p,t}), g(x_{p',t'}))$ and a positive semi-definite matrix K_{task} for learning inter-task similarities. Specifically, we define the multi-task kernel K_m as follows

$$k_m([x_{p,t}, \mathfrak{t}], [x_{p',t'}, \mathfrak{t}']) = k(g(x_{p,t}), g(x_{p',t'})) * k_{\text{task}}(\mathfrak{t}, \mathfrak{t}') \quad (6)$$

The GP prior will be defined by replacing the task specific kernel $K^{\mathfrak{t}}$ in the equation 4 with the multi-task kernel K_m. We use the optimization steps similar to DGP and the inference is done by using the standard GP formulae.

Relating MDGPR to equation 1, the global parameters Θ are the parameters of the deep network g, and the task specific parameter Φ is the positive semi-definite matrix K_{task}.

Model Agnostic Meta Learning (MAML): MAML (Finn et al., 2017) is a popular meta learning algorithm that can be used to quickly adapt Deep Neural Networks on new tasks in a few-shot setting. In MAML, the set of initialization parameters for the neural network are explicitly learned such that the network can generalize well on a new task with a small number of gradient steps and training samples.

Relating to equation 1, the global parameters Θ can be considered as the initial set of parameters for the neural network that are learned and shared across all the tasks. Task specific parameters Φ are adapted from Θ by taking K gradient steps using the task's performance data.

For evaluating a task $\mathfrak{t}$, we consider rest of the tasks in our dataset as helpers ($\mathfrak{t}' \in \mathfrak{T} - \{\mathfrak{t}\}$) and use them to train the initial set of parameters Θ. The initial parameters are then updated by fine-tuning the network on the training set for $\mathfrak{t}$ using gradient descent.

A.2 Comparison between mBERT and XLMR Tokenizers

The FERT and PCW metrics as proposed by Rust et al. (2021), have been compared for mBERT and XLMR in figure 4. As can be seen, for most languages the metric values are similar across the two tokenizers, however for languages like Chinese and Japanese, there is a dramatic increase in the values for XLMR. Interestingly, when we compare the zero-shot performance between mBERT and XLMR on structure prediction tasks like UDPOS and WikiANN, we see a surprisingly large drop (upto 20% absolute drop) in the performance for XLMR on these both Chinese and Japanese, whereas usually XLMR outperforms mBERT on these tasks (Refer to figure 5). This observation along with the feature importance for the tokenizer features that we observed for Group Lasso (3) indicate that tokenizer quality might play some role in the zero-shot transfer capabilities of the multilingual models.

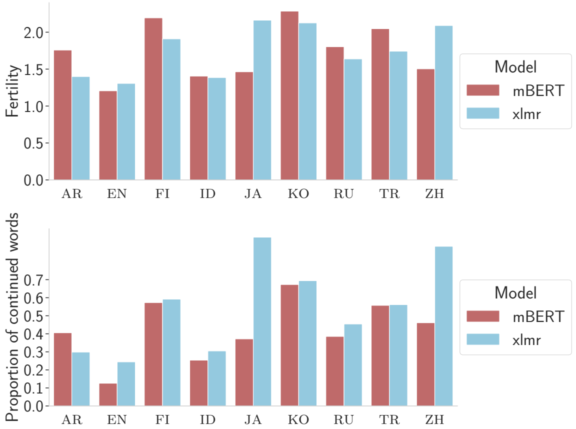
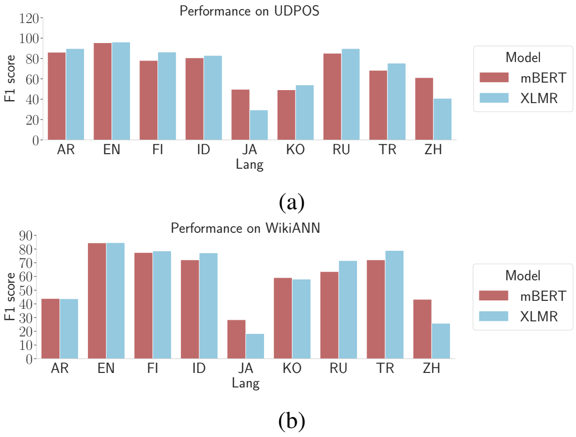


Figure 4: Comparison of Tokenizer metrics as described by Rust et al. (2021) on different languages for MBERT and XLMR. For most languages both model's have similar values of fertility and proportion of continued words, however for Chinese and Japanese the values for XLMR are much higher, which might indicate the subpar quality of XLMR's tokenizer in these languages.



(a)

(b)

Figure 5: Zero-shot performance comparison between mBERT and XLMR on (a) UDPOS and (b) WikiANN (NER) tasks, as given in Ruder et al. (2021)

Task	$\lvert\mathcal{T}\rvert$	Baselines		Single Task Models			Multi Task Models			
		Average within Task	Average across Tasks	Lasso	XGBoost	DGPR	Group Lasso	CMF	MDGPR	MAML
MLQA	7	4.87	4.59	6.39	7.47	6.12	3.45	3.18	**2.42**	3.75
PAWS	7	4.01	2.96	3.97	3.01	3.53	2.34	2.75	**1.92**	6.77
XCOPA	8	3.44	3.63	3.54	4.24	3.10	3.30	2.86	**2.59**	5.38
TyDiQA	9	5.06	7.08	**3.42**	6.44	3.94	5.09	4.59	3.92	8.34
XQUAD	10	6.56	2.97	**2.89**	4.69	3.26	4.16	4.37	3.13	4.86
LAReQA	10	5.57	2.79	2.59	4.40	2.64	2.22	1.96	**1.75**	8.74
MewsliX	10	19.23	15.48	12.15	15.54	17.52	10.53	**9.54**	15.99	14.72
XNLI	14	5.29	2.94	3.29	**2.60**	2.95	3.18	3.89	2.98	5.05
WikiANN	32	14.79	10.54	9.37	11.13	11.51	10.30	8.91	**8.62**	11.80
Tatoeba	35	14.63	11.86	**6.43**	9.57	6.38	6.46	7.21	**6.16**	12.13
UDPOS	48	12.10	7.43	7.05	6.37	**6.18**	8.94	**6.58**	6.87	7.97
Average	19	8.69	6.57	5.55	6.86	6.10	5.45	**5.08**	5.12	8.14
Average ($\lvert\mathcal{T}\rvert \leq 10$)	9	6.96	5.64	4.99	6.54	5.73	4.44	**4.18**	4.53	7.51

Table 2: Mean Absolute Errors (Scaled by 100 for readability) for different models trained to predict the zero shot performance of mBERT. In the "Average" row we average the MAEs across all the tasks and in the "Average Low" Res Tasks", we consider the tasks with fewer than 10 target languages and take the average of the MAEs for those tasks.

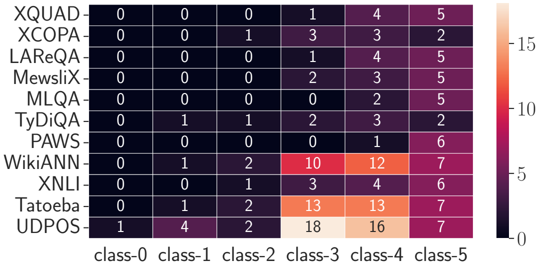


Figure 6: Class wise distribution of languages for different tasks. Languages have been categorized based on the taxonomy provided by Joshi et al. (2020)

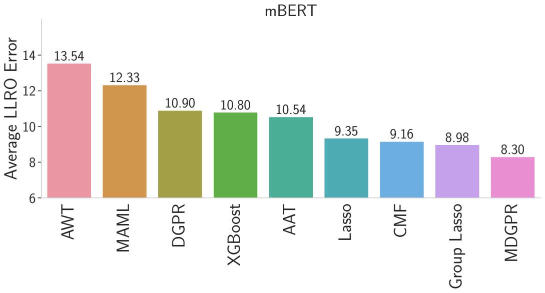


Figure 7: Leave Low Resource Out (LLRO) results for mBERT

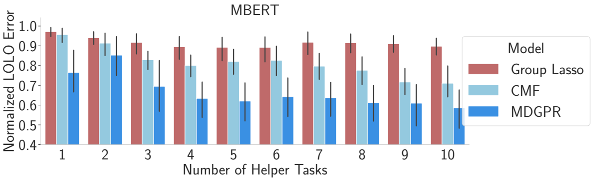


Figure 8: Number of helper tasks vs. LOLO MAE for mBERT. Errors for different model types (Group Lasso, CMF and MDGPR) and tasks are scaled by diving them by the maximum error value.

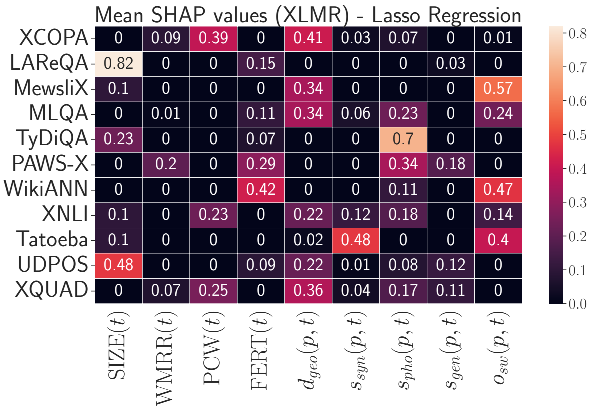


Figure 9: Task-wise mean SHAP values of different features for the Single Task Lasso Regression model trained on XLMR zero-shot performance data.

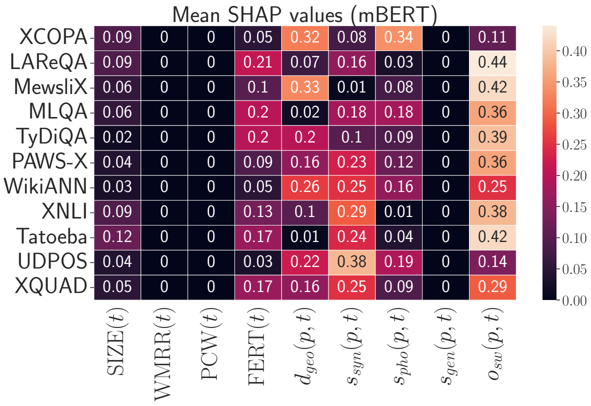


Figure 10: Task-wise mean SHAP values of different features for the Group Lasso model trained on mBERT zero-shot performance data.

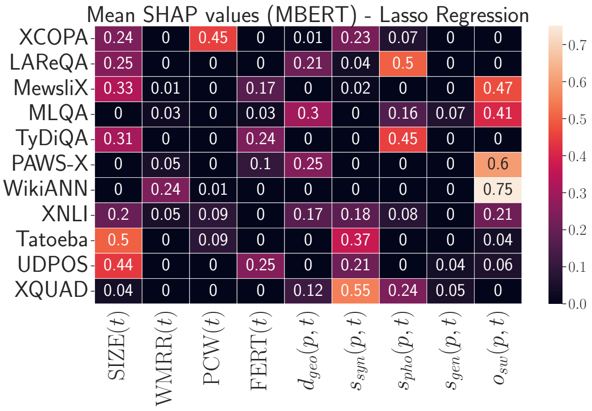


Figure 11: Task-wise mean SHAP values of different features for the Single Task Lasso Regression model trained on mBERT zero-shot performance data.