Model Selection for Cross-Lingual Transfer

Yang Chen and **Alan Ritter**
School of Interactive Computing
Georgia Institute of Technology
`{yang.chen, alan.ritter}@cc.gatech.edu`

Abstract

Transformers that are pre-trained on multilingual corpora, such as, mBERT and XLM-RoBERTa, have achieved impressive cross-lingual transfer capabilities. In the zero-shot transfer setting, only English training data is used, and the fine-tuned model is evaluated on another target language. While this works surprisingly well, substantial variance has been observed in target language performance between different fine-tuning runs, and in the zero-shot setup, no target-language development data is available to select among multiple fine-tuned models. Prior work has relied on English dev data to select among models that are fine-tuned with different learning rates, number of steps and other hyperparameters, often resulting in suboptimal choices. In this paper, we show that it is possible to select consistently better models when small amounts of annotated data are available in auxiliary pivot languages. We propose a machine learning approach to model selection that uses the fine-tuned model's own internal representations to predict its cross-lingual capabilities. In extensive experiments we find that this method consistently selects better models than English validation data across twenty five languages (including eight low-resource languages), and often achieves results that are comparable to model selection using target language development data.[1]

1 Introduction

Pre-trained Transformers (Vaswani et al., 2017; Devlin et al., 2019) have achieved state-of-the-art results on a range of NLP tasks, often approaching human inter-rater agreement (Joshi et al., 2020a). These models have also been demonstrated to learn effective cross-lingual representations, even without access to parallel text or bilingual lexicons (Wu and Dredze, 2019; Pires et al., 2019).

In the zero-shot transfer learning, training and development data are only assumed in a high resource source language (e.g. English), and performance is evaluated on another target language. Because no target language annotations are assumed, source language data is typically used to select among models that are fine-tuned with different hyperparameters and random seeds. However, recent work has shown that English dev accuracy does not always correlate well with target language performance (Keung et al., 2020).

In this paper, we propose an alternative strategy for model selection in zero-shot transfer. Our approach, dubbed Learned Model Selection (LMS), learns a function that scores the compatibility between a fine-tuned multilingual Transformer, and a target language. The compatibility score is calculated based on features of the multilingual model's learned representations. This is done by aggregating representations over an unlabeled target language text corpus after fine-tuning on source language data. We show that these model-specific features effectively capture information about how the cross-lingual representations will transfer. We also make use of language embeddings from the `lang2vec` package (Malaviya et al., 2017),[2] which have been shown to encode typological information that may help inform how a multilingual model will transfer to a particular target. These model and language features are combined in a bilinear layer to compute a ranking on the fine-tuned models. Parameters of the ranking function are optimized to minimize a pairwise loss on a set of held-out models, using one or more auxiliary pivot languages. Our method assumes training data in English, in addition to small amounts of auxiliary language data. This corresponds to a scenario where the multilingual model needs to be quickly adapted to a new language. LMS does not rely on any annotated data in the target language, yet it is

---

[1] Our code and data is available at: `https://github.com/edchengg/model_selection`

[2] `https://github.com/antonisa/lang2vec`

(a) English Development Selection
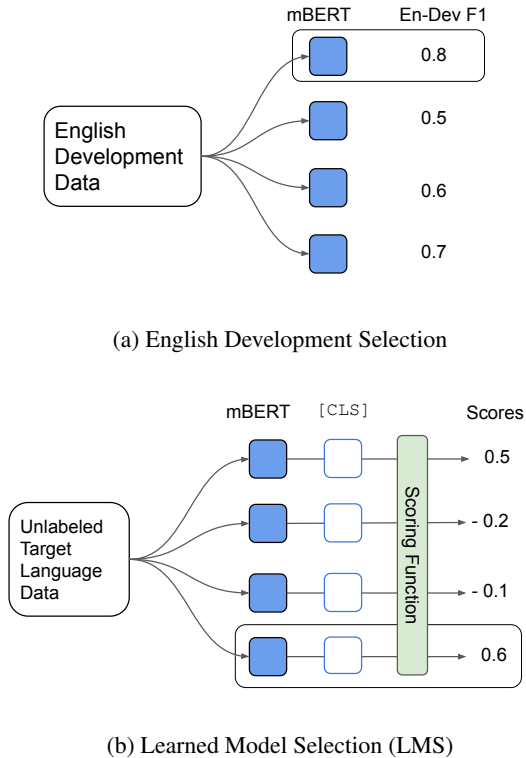

(b) Learned Model Selection (LMS)

Figure 1: An illustration of our approach to select the best model for zero-shot cross-lingual transfer. (a) Prior work selects the best model using source language development data. (b) LMS: A learned function scores fine-tuned models based on their hidden layer representations when encoding unlabeled target language data.

effective in learning to predict how well fine-tuned representations will transfer.

In experiments on twenty five languages, LMS consistently selects models with better target-language performance than those chosen using English dev data. Furthermore, our proposed approach improves performance on low-resource languages such as Quechua, Maori and Turkmen that are not included in the pretraining corpus (§6.1).

2   Background: Cross-Lingual Transfer Learning

The zero-shot setting considered in this paper works as follows. A Transformer model is first pre-trained using a standard masked language model objective. The only difference from the mono-lingual approach to contextual word representations (Peters et al., 2018; Devlin et al., 2019) is the pre-training corpus, which contains text written in multiple languages; for example, mBERT is trained on Wikipedia data from 104 languages.

After pre-training, the resulting network encodes language-independent representations that support surprisingly effective cross-lingual transfer, simply by fine-tuning with English data. For example, after fine-tuning mBERT using the English portion of the CoNLL Named Entity Recognition dataset, the resulting model can perform inference directly on Spanish text, achieving an F_1 score around 75, and outperforming prior work using cross-lingual word embeddings (Xie et al., 2018; Mikolov et al., 2013). A challenge, however, is the relatively high variance across multiple training runs. Although mean F_1 on Spanish is 75, the performance of 60 fine-tuned models with different learning rates and random seeds ranges from around 70 F_1 to 78. In zero-shot learning, no validation/development data is available in the target language, motivating the need for a machine learning approach to model selection.

3   Ranking Model Compatibility with a Target Language

Given a set of multilingual BERT-based models, $M = m_1, m_2, ..., m_n$ that are fine-tuned on an English training set using different hyperparameters and random seeds, our goal is to select the model that performs best on a specific target language, l_{target}. Our approach (LMS) learns to rank a set of models based on two sources of information: (1) the models' own internal representations, and (2) lang2vec representations of the target language (Malaviya et al., 2017).

We adopt a pairwise approach to learning to rank (Burges et al., 2005; Köppel et al., 2019). The learned ranking is computed using a scoring function, $s(m, l) = f(g_{\text{mBERT}}(m), g_{\text{lang2vec}}(l))$, where $g_{\text{mBERT}}(m)$ is a feature vector for model m, which is computed from the model's own hidden state representations, and $g_{\text{lang2vec}}(l)$ is the lang2vec representation of language l. The model and language features are each passed through a feed-forward neural network and then combined using a bilinear layer to calculate a final score as follows:

$$
\begin{aligned}
s(m, l) &= f(g_{\text{mBERT}}(m), g_{\text{lang2vec}}(l)) \\
&= \text{FFNN}(g_{\text{mBERT}}(m))^T W_{\text{bi}} \text{FFNN}(g_{\text{lang2vec}}(l))
\end{aligned}
$$

Using the above score, we can represent the probability that model m_i performs better than m_j on language l:

$$
P(m_i \rhd_l m_j) = \sigma(s(m_i, l) - s(m_j, l))
$$

where $\sigma(\cdot)$ is the sigmoid function. To tune the parameters of the scoring function, which include the feed-forward and bilinear layers, we minimize cross-entropy loss:

$$C = \sum_{l \in L \setminus \{l_{\text{target}}\}} \sum_{m_i \in M} \sum_{m_j \in M} -C_{m_i, m_j, l} \quad (1)$$

where

$$
\begin{aligned}
C_{m_i, m_j, l} = \quad & \mathbb{1}[m_i \triangleright_l m_j] \log P(m_i \triangleright_l m_j) \\
& + \mathbb{1}[m_j \triangleright_l m_i] \log P(m_j \triangleright_l m_i)
\end{aligned}
$$

Here $\mathbb{1}[m_j \triangleright_l m_i]$ is an indicator function that has the value 1 if m_j outperforms m_i, as evaluated using labeled development data in language l.

The first sum in Equation 1 ranges over all languages where development data is available (this excludes the target language). After tuning parameters to minimize cross-entropy loss on these languages, models are ranked based on their scores for the target language, and the highest scoring model, $\hat{m} = \arg\max_m s(m, l_{\text{target}})$, is selected.

4 Tasks and Datasets

We perform model selection experiments on five well-studied NLP tasks in the zero-shot transfer setting: part-of-speech (POS) tagging, question answering (QA), relation extraction (RE), event-argument role labeling (ARL), and named entity recognition (NER). In total, we cover 25 target languages including 8 low-resource languages in our experiments following prior work (shown in Table 1). We adopt the best performing model from Soares et al. (2019), [ENTITY MARKERS - ENTITY START], for RE and ARL. For other tasks, we use established task-specific layers and evaluation protocols, following the references in Table 1. Labeled training data for each task is assumed in English and trained models are evaluated on each target language.

4.1 Low-resource Languages

To evaluate LMS on truly low-resource languages, we use the 8 target languages (summarized in Table 2) following (Pfeiffer et al., 2020; Xia et al., 2021) which uses the WikiAnn NER dataset (Pan et al., 2017). These languages are considered low-resource because: 1) the Wikipedia size ranges from 4k to 22k; 2) they are not covered by pre-trained multilingual models (i.e., by mBERT and XLM-RoBERTa). The train, development, and test

partition of Rahimi et al. (2019) is used following the XTREME benchmark's NER setup (Hu et al., 2020). The related language used for the Pivot-Dev baseline is chosen following Xia et al. (2021), which is based on LangRank (Lin et al., 2019).

5 Experimental Design

For a multilingual NLP task with n languages $L : \{l_1, ..., l_n\}$, our goal is to select the model that performs best on a new target language, $l_{\text{target}} \notin L$. We assume the available resources are English training and development data, in addition to a small development set for each of the pivot languages, L. First, a set of mBERT models, M, are fine-tuned on an English training set using different hyperparameters and random seeds and shuffled into meta-train/dev/test sets. We then evaluate each model, m_i, on the pivot languages' dev sets to calculate a set of gold rankings, $\triangleright_l$, that are used in the cross-entropy loss (Equation 1). Model-specific features are extracted from the fine-tuned mBERTs, by feeding unlabeled pivot language text as input. **Development and Evaluation** mBERT models in the meta-dev set are used to experiment with different model and language features. Evaluation is performed using models in the meta-test set. We use the leave-one-language-out setup for each task during evaluation. For each target language, we rank models using the learned scoring function, select the highest scoring model, and report results in Table 3.

5.1 Baselines and Oracles

En-Dev is our main baseline following standard practice for model selection in zero-shot transfer learning (Wu and Dredze, 2019; Pires et al., 2019). Because our approach assumes additional development data in auxiliary languages, we also include a baseline that uses pivot-language language dev data.[3] In addition, we compare against an oracle that selects models using 100 annotated sentences from the target language dev set to examine how our approach compares with the more costly alternative of annotating small amounts of target language data. Finally, we include an oracle that simply picks the best model using the full target language development set (All-Target). All baselines and oracles are summarized below:

---

[3]The auxiliary language with highest similarity to the target language, as measured using cosine similarity between `lang2vec` embeddings, is used in this baseline.

Task	Dataset	References	Target Languages
POS	UD (Nivre et al., 2016)	Wu and Dredze (2019)	*ar,bg,da,de,es,fa,hu,it nl,pt,ro,sk,sl,sv,vi,zh*
QA	MLQA (Lewis et al., 2020)	Lewis et al. (2020)	*ar, de, es, hi, vi, zh*
RE	ACE05 (Walker et al., 2006)	Subburathinam et al. (2019)	*ar, zh*
ARL	ACE05 (Walker et al., 2006)	Subburathinam et al. (2019)	*ar, zh*
NER	CoNLL (Sang, 2002)	Wu and Dredze (2019)	*de, es, nl, zh*
Low-resource languages			
NER	WikiAnn (Pan et al., 2017)	Pfeiffer et al. (2020); Xia et al. (2021)	*cdo, gn, ilo, mhr mi, tk, qu, xmf*

Table 1: 25 target languages and five tasks used in our experiments. English is used as the source language. ar: Arabic, bg: Bulgarian, da: Danish, de: German, es: Spanish, fa: Persian, hi: Hindi, hu: Hungarian, it: Italian, nl: Dutch, pt: Portuguese, ro: Romanian, sk: Slovak, sl: Slovene, sv: Swedish, vi: Vietnamese, zh: Chinese. Low-resource languages information can be found in Table 2.

Language	Code	Language Family	Related Language	#Wiki articles
Min Dong	cdo	Sino-Tibetan	Chinese (zh)	15k
Guarani	gn	Tupian	Spanish (es)	4k
Ilocano	ilo	Austronesian	Indonesian (id)	14k
Meadow Mari	mhr	Uralic	Russian (ru)	10k
Maori	mi	Austronesian	Indonesian (id)	7k
Turkmen	tk	Turkic	Turkish (tr)	6k
Quechua	qu	Quechua	Spanish (es)	22k
Mingrelian	xmf	Kartvelian	Georgian (ka)	13k

Table 2: Low-resource target languages used in the WikiAnn NER task. Related languages, used in the Pivot-Dev baseline, are selected following (Xia et al., 2021).

- **En-Dev (baseline):** chooses the fine-tuned mBERT with best performance on the English dev set.

- **Pivot-Dev (baseline):** chooses the fine-tuned mBERT with best performance on development data in the most similar pivot language (similarity to the target language is measured using `lang2vec` embeddings).

- **100-Target (oracle):** chooses the fine-tuned mBERT with best performance on 100 labeled target language instances.

- **All-Target (oracle):** chooses the fine-tuned mBERT using the full target language dev set.

5.2 Hyperparameters and Other Settings

To train the scoring function, $s(\cdot)$, we use Adam (Kingma and Ba, 2015), and select the batch size among $\{16, 32, 64, 128\}$, learning rate λ among $\{1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 1 \times 10^{-6}\}$, and train for $\{3\}$ epochs. The

scoring function, $s(\cdot)$, contains a 2-layer FFNN with 1024 hidden units and ReLU activation (Glorot et al., 2011). The base cased mBERT has 179M parameters and a vocabulary of around 120k wordpieces. Both the pre-trained Transformer layers and task-specific layers are fine-tuned using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and an L2 weight decay of 0.01. Model candidates are fine-tuned with varying learning rates and number of epochs with the following settings: learning rate $\in \{3 \times 10^{-5}, 5 \times 10^{-5}, 7 \times 10^{-5}\}$; number of epochs $\in \{3, 4, 5, 6\}$; batch size $\in \{32\}$; random seeds $\in \{0, 1, ..., 239\}$. 240 mBERT models with different random seeds are fine-tuned with 12 different hyperparameter settings (20 random seeds for each set of hyperparameters), and then split into meta-train/dev/test sets (120/60/60). All models are trained on an RTX 2080 Ti.

6 Evaluation

Below we report model selection results on mBERTs in the meta-test set for each of the five tasks.

POS Table 3 presents POS accuracies on the test set, using various approaches to model selection for the fifteen target languages. LMS outperforms En-Dev and Pivot-Dev except in the case of Swedish (sv) and Dutch (nl). Interestingly, model selection for Italian with Spanish dev set does not outperform LMS. We use (Wu and Dredze, 2019) as references for zero-shot cross-lingual transfer with mBERT.

QA Our method selects a model with higher F1 across all languages compared with En-Dev, although we find that Pivot-Dev performs slightly

Task	Lang	Ref	En-Dev	Pivot-Dev	LMS	100-Target	All-Target	# All-Target
POS (Acc)	ar	-	49.7	50.3 (de)**	**51.6****	50.6	52.7	786
	de	89.8	89.3	88.7 (nl)	**89.8****	89.4	89.9	799
	es	85.2	84.8	85.3 (nl)**	**85.6****	84.8	85.1	1552
	nl	75.9	75.7	**75.9** (de)**	75.9**	75.5	76.0	349
	zh	-	66.9	66.9 (de)**	**68.0****	67.3	68.8	500
	bg	87.4	87.1	87.0 (es)	**87.9****	87.9	87.9	1115
	da	88.3	88.6	88.8 (nl)**	**88.9****	88.6	89.2	322
	fa	72.8	71.6	71.6 (es)	**73.6****	73.6	73.7	599
	hu	83.2	82.5	82.0 (de)	**83.3****	83.3	83.1	179
	it	84.7	84.5	84.9 (es)**	**85.2****	85.4	85.8	489
	pt	82.1	81.8	81.8 (es)	**82.2****	81.8	82.2	271
	ro	84.7	83.8	84.2 (es)**	**84.7****	84.4	85.4	1191
	sk	83.6	83.7	83.6 (es)	**84.2****	83.6	84.8	1060
	sl	84.2	84.5	83.6 (es)	**85.2****	83.8	85.5	735
	sv	91.3	91.4	**91.8** (nl)**	91.7**	91.3	91.8	504
	AVG En-Dev Δ	-	0.0	0.0	**0.9**	0.3	1.0	-
QA (F₁)	ar	45.7	47.7	**49.4** (de)**	49.3**	49.4	49.4	517
	de	57.9	55.3	55.8 (ar)**	**55.9****	57.1	55.8	512
	es	64.3	64.9	64.7 (ar)	**65.0**	64.5	65.1	500
	zh	57.5	58.0	58.0 (de)	**58.1**	58.1	58.4	504
	hi	43.8	39.1	42.1 (es)**	**42.4****	38.8	42.9	507
	vi	57.1	57.3	56.9 (ar)	**58.2****	59.1	58.1	511
	AVG En-Dev Δ	-	0.0	0.8	**1.1**	0.8	1.2	-
RE (F₁)	ar	39.4	36.1	35.3 (zh)	**39.5****	34.7	41.9	4482
	zh	32.7	67.7	67.4 (ar)	**70.8****	68.2	69.1	7096
	AVG En-Dev Δ	-	0.0	-0.6	**3.3**	-0.5	3.6	-
ARL (F₁)	ar	16.5	44.1	**48.1** (zh)**	47.1**	44.1	47.2	1221
	zh	23.5	61.0	61.3 (ar)	**62.1****	62.5	63.8	2226
	AVG En-Dev Δ	-	0.0	**2.2**	2.1	0.8	3.0	-
NER_CoNLL (F₁)	de	69.6	69.9	70.7 (nl)**	**71.0****	66.7	72.1	2867
	es	75.0	74.6	73.1 (nl)	**75.7****	75.7	75.7	1915
	nl	77.6	78.7	**79.3** (de)**	78.9**	78.7	80.3	2895
	zh	51.9	54.9	53.0 (de)	**55.1**	55.4	56.9	4499
	AVG En-Dev Δ	-	0.0	-0.5	**1.1**	-1.1	2.1	-
Low-resource languages								
NER_WikiAnn (F₁)	cdo	14.2	11.0	12.4 (zh)	**19.4****	19.4	19.4	100
	gn	45.4	45.0	47.1 (es)	**49.0****	46.2	46.2	100
	ilo	63.5	**61.4**	59.2 (id)	61.1	66.1	66.1	100
	mhr	46.0	45.3	41.1 (ru)	**48.6****	48.3	48.3	100
	mi	21.8	28.2	33.8 (id)**	**43.7****	55.6	55.6	100
	tk	47.2	51.6	**55.5** (tr)**	54.9**	56.5	56.5	100
	qu	54.9	59.8	**62.2** (es)**	59.8	63.4	63.4	100
	xmf	31.1	34.3	36.4 (ka)**	**37.5****	37.8	37.8	100
	AVG En-Dev Δ	-	0.0	1.4	**4.7**	7.1	7.1	-

Table 3: Model scores selected based on LMS for POS, QA, RE, ARL, and NER. All mBERT models are fine-tuned on English training data. En-Dev / Pivot-Dev / 100-Target / All-Target: model selection based on the highest F1 of English dev set / Pivot language dev set (pivot language in bracket) / 100 target language dev set examples / target language dev set. LMS: model selection based on the highest scores for the target language: $\arg\max_m s(m, l_{\text{target}})$; "# All-Target" is the number of labeled target-language sentences used for model selection in the All-Target oracle. Bold / underlined indicates the best / second best. AVG En-DevΔ: average differences with En-Dev baseline. Significance compared to the En-Dev is indicated with **($p < 0.05$) – all tests are computed using the paired bootstrap procedure (Berg-Kirkpatrick et al., 2012).

better on Arabic (ar). We use (Lewis et al., 2020) as references for zero-shot cross-lingual transfer with mBERT.

ARL and RE In Table 3, our method selects models with higher F1 scores compared to En-Dev. It also outperforms 100-Target on Arabic. We hypoth-

esize this is because 100 target-language examples is not sufficient for effective model selection, as the dataset contains a large proportion of negative examples (no relation). Also, RE and ARL have large label sizes (18 and 35) so a random sample of 100 instances might not cover every label. In contrast, the full dev set contains thousands of examples. We use a Graph Convolutional Network (GCN) (Subburathinam et al., 2019) as a reference (see Appendix A.1 for details) and models were selected using the English dev set.

CoNLL NER As illustrated in Table 3, our method selects models with a higher F1 score than En-Dev. Besides, it outperforms model selection using small amounts of target-language annotations (100-Target) on Dutch (nl) and German (de) and selects a model that performs as well on Spanish (es). On average, LMS achieves 1.6 point increase in F1 score relative to Pivot-Dev. We use (Wu and Dredze, 2019) as references for zero-shot cross-lingual transfer with mBERT.

Model Score Distributions Figure 2 visualizes the En-Dev and LMS results on the test set in the context of the score distributions of the 60 models in the meta-test set, using kernel density estimation. English development data tends to select models that perform only slightly better than average, whereas LMS does significantly better.
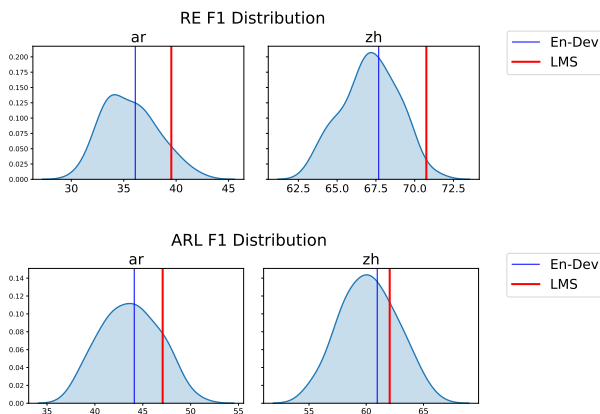

Figure 2: Model F1 score distributions for RE and ARL. Red line: LMS and blue line: En-Dev. X-axis is F1 score. Selecting models with LMS achieve better results compared to En-Dev.

6.1 Evaluation on low-resource languages

We present results of low-resource languages NER in the bottom section of Table 3, where we use 40 pivot languages in the XTREME benchmark (Hu et al., 2020) to train LMS and test on 8 target lan-

guages. LMS, outperforms the En-Dev and Pivot-Dev baselines, leading to an average gain of 4.7 and 3.3 F1 respectively. Since the setting is targeting truly low-resource languages where `lang2vec` might not be available, the scoring function thus directly predicts a score based on the representation from unlabeled target language text. We use the mBERT zero-shot cross-lingual transfer results from (Pfeiffer et al., 2020) as references.

6.2 Evaluation on multilingual fine-tuned models

An interesting question is whether fine-tuning on available development data in the auxiliary languages can improve performance. Since our model assumes access to small amounts of labeled data in a set of pivot languages, we experiment with multilingual fine-tuning and show LMS is still beneficial for selecting among models that are fine-tuned on both English and pivot language data.

Part of speech tagging experiments are presented in Table 4, where all mBERT models are fine-tuned on English and the development sets of five pivot languages (ar,de,es,nl,zh). A single LMS is then trained using English fine-tuning, with gold rankings computed on the pivot languages. Then, we directly apply the English LMS model to do model selection on the multilingual fine-tuned mBERT models. We find LMS outperforms the En-Dev baseline on seven out of the ten target languages used in our evaluation, with an average gain of 0.1 accuracy. This also demonstrates LMS that is trained on English fine-tuned representations generalizes to multilingual fine-tuning. We use models that are fine-tuned only on English data, from Wu and Dredze (2019), as references, and find that multilingual fine-tuning shows better cross-lingual transfer performance compared to fine-tuning on only English data.

7 Analysis

In Section 6, we empirically demonstrated that our learned scoring function, $s(\cdot)$, consistently selects better models than the standard approach (En-Dev), and is comparable to small amounts of labeled target language data. Section 7.1 presents additional analysis of our approach, exploring the impact of various modeling choices with $\{ar, de, es, nl, zh\}$. In addition, analysis of generalization beyond mBERT and across tasks capability are present in Appendices A.3 and A.4.

Task	Lang	Ref	En-Dev	Pivot-Dev	LMS	100-Target	All-Target	# All-Target
POS (Acc)	bg	87.4	90.3	90.3 (es)	**90.8**[**]	90.6	90.8	1115
	da	88.3	**89.4**	89.2 (nl)	89.3	88.6	89.2	322
	fa	72.8	80.8	79.9 (es)	**81.1**[**]	89.4	89.7	599
	hu	83.2	**84.2**	84.1 (de)	84.0	82.1	82.1	179
	it	84.7	94.0	94.0 (es)	**94.9**[**]	95.1	95.1	489
	pt	82.1	91.1	91.3 (es)	91.1	91.1	91.5	271
	ro	84.7	88.7	88.7 (es)	**88.8**[**]	89.1	89.1	1191
	sk	83.6	88.3	88.3 (es)	**88.6**[**]	89.0	89.0	1060
	sl	84.2	86.1	86.4 (es)	**86.8**[**]	86.8	86.8	735
	sv	91.3	91.8	90.9 (nl)	**92.2**[**]	92.5	92.5	504
	AVG En-Dev Δ	-	0.0	0.0	**0.1**	0.6	0.7	-

Table 4: Model scores selected based on LMS for POS. En-Dev / Pivot-Dev / 100-Target / All-Target: model selection based on the highest F1 of English dev set / Pivot language dev set (pivot language in bracket) / 100 target language dev set examples / target language dev set. All mBERT models are fine-tuned on English training data in addition with a small amount of data from pivot languages in $\{ar, de, es, nl, zh\}$. We use English data fine-tuned mBERT results from Wu and Dredze (2019) as references (Ref). Significance compared to the En-Dev is indicated with [**]$(p < 0.05)$ – all tests are computed using the paired bootstrap procedure (Berg-Kirkpatrick et al., 2012).

7.1 Model and Language Features

This section explores the impact of different choices for model and language representations for LMS. Four types of model features and two language embeddings are explored. We start by delineating possible choices for representations, then describe the details of our experiments, results, and the final choices used in §6.

Four model-specific features are described below. Note `[CLS]` vectors are extracted from mBERT by feeding unlabeled text as input.

- `[Eng]`: Averaged `[CLS]` vectors computed over an unlabeled English text corpus are used for both training and testing.[4]

- `[Pivot]`: During training, `[CLS]` vectors are averaged over an unlabeled text corpus in the pivot language. At test time, `[CLS]` embeddings are averaged over an unlabeled corpus in the target language. We use the target-language development set (ignoring labels) for this purpose in our experiments.

- `[Target]`: `[CLS]` vectors are averaged over a text corpus in the target language (for both training and testing).

- Fusion: A linear combination of the above features. Weights on each representation are learned during training.

Two types of language embeddings are examined.

- `lang2vec`: 512-dimensional vectors learned by a neural network trained for typological prediction (Malaviya et al., 2017).

- `syntax`: 103-dimensional binary vectors, which capture syntax features from the URIEL knowledge base (Littell et al., 2017).

First, we determine the choice of model-specific features by averaging performance across both language embeddings. Table 5 reports averaged evaluation metrics for each model-specific representation across all target languages with En-Dev as a baseline.

Averaged evaluation metrics across all target languages for each language embedding are reported in Table 6. In addition to evaluating the effectiveness of each language embedding, we also experimented with a variant of our scoring function that does not include any language embeddings as input. Results are reported on mBERT models in the meta-dev set and the target languages' dev sets for all experiments in this section.

In Table 5, `[PIVOT]` features achieve top-2 performance in all five tasks. `[Eng]` and `[Target]` achieve mixed results, and the fusion of three features does not effectively incorporate the advantages of each representation, except in the case of ARL. Table 6 shows that `lang2vec` outperforms `syntax` for all tasks but ARL and also outperforms our approach when language embeddings are not included. Thus, `lang2vec` and `[PIVOT]` are used for all experiments in Section 6.

---

[4]In our experiments, sentences in the English dev set are used for this purpose (ignoring the labels).

Task	En-Dev	[Eng]	[Pivot]	[Target]	Fusion
POS	74.69	**75.58**	<u>75.54</u>	75.48	75.04
QA	56.31	56.49	**56.79**	<u>56.68</u>	56.63
RE	51.81	54.92	**55.57**	<u>55.56</u>	54.57
ARL	50.98	51.99	<u>53.74</u>	52.31	**54.69**
NER	70.45	70.64	<u>71.18</u>	**71.87**	70.66
Avg	60.85	61.92	**62.60**	<u>62.38</u>	62.32

Table 5: Model-specific feature analysis. We use mBERT models in the meta-dev set for analysis. Each number represents average of scores across all the target languages in a particular task.

Task	En-Dev	lang2vec	syntax	None
POS	74.69	**75.72**	75.36	75.20
QA	56.31	**56.81**	56.77	56.49
RE	51.81	**55.92**	55.22	52.53
ARL	50.98	53.60	**53.88**	53.14
NER	70.45	**71.37**	70.98	70.08
Avg	60.85	**62.68**	62.44	61.49

Table 6: Language embedding analysis across lang2vec, syntax, and no language embedding. We use mBERT models in the meta-dev set for analysis. Each number represents average of scores across all the target languages in a particular task.

8 Related Work

Recent work has explored hyper-parameter optimization (Klein et al., 2019), and model selection for a new task. task2vec (Achille et al., 2019) presents a meta-learning approach to selecting a pre-trained feature extractor from a library for a new visual task. More concretely, task2vec represents tasks in a vector space and is capable of predicting task similarities and taxonomic relations. It encodes a new task and selects the best feature extractor trained on the most similar task. Unlike task2vec, we select a trained model for a specific task, and we represent a trained model with model-specific features on a target language.

MAML (Finn et al., 2017; Rajeswaran et al., 2019) is another approach to meta-learning, pre-training a single model with a meta-loss to initialize a set of parameters that can be quickly fine-tuned for related tasks. Nooralahzadeh et al. (2020) explore the use of MAML in the cross-lingual transfer setting. MAML is designed to support few-shot learning through better initialization of model parameters and does not address the problem of model selection. In contrast, our approach improves model selection in the zero-shot cross-lingual transfer setting.

Most relevant to our work, Xia et al. (2020) use regression methods to predict a model's performance on an NLP task. They formulate this as a regression problem based on features of the task (dataset size, average sentence length, etc.), incorporating a discrete feature to represent the choice of model. In contrast, LMS inspects a model's internal representations, thus it is suitable for predicting which out of a set of fine-tuned models will best transfer to a target language. Also relevant is prior work on learning to select the best language to transfer from (Lin et al., 2019).

There is a need for more NLP research on low-resource languages (Joshi et al., 2020b). Lauscher et al. (2020) present a number of challenges in transferring to languages with few resources using pre-trained Transformers. Our experiments do cover a set of 8 truly low-resource languages following prior work (Pfeiffer et al., 2020; Xia et al., 2021) and a fairly diverse set of languages, including Arabic and Chinese. We believe that there is still a need for more research on multilingual NLP for high-resource languages as well, as this is not a solved problem. Finally, we note that there are several other prominent benchmarks for evaluating cross-lingual transfer including XTERME (Hu et al., 2020) and XGLUE (Liang et al., 2020), both of which include some datasets used in this work.

9 Conclusion

In this paper, we presented a machine learning approach to model selection for zero-shot cross-lingual transfer, which is appropriate when small amounts of development data are available in one or more pivot languages, but not in the target language. We showed that our approach improves over the standard practice of model selection using source language development data. Experiments on five well-studied NLP tasks show that by inspecting internal representations, our method consistently selects better models. LMS also achieves comparable results to the more expensive alternative of annotating small amounts of target-language development data. Finally, we demonstrated that LMS selects better models for low-resource languages, such as Quechua and Maori, that are not included during pretraining.

Acknowledgements

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *Proceedings of the International Conference on Computer Vision*.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of Empirical Methods in Natural Language Processing*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. In *Transactions of the Association for Computational Linguistics*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of International Conference on Machine Learning*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of Artificial Intelligence and Statistics*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A benchmark for evaluating cross-lingual generalization. In *Proceedings of International Conference on Machine Learning*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020a. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020b. The state and fate of linguistic diversity and inclusion in the NLP world. *Proceedings of the Association for Computational Linguistics*.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. Don't use english dev: On the zero-shot cross-lingual evaluation of contextual embeddings. *Proceedings of Empirical Methods in Natural Language Processing*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representation*.

Aaron Klein, Zhenwen Dai, Frank Hutter, Neil Lawrence, and Javier Gonzalez. 2019. Meta-surrogate benchmarking for hyperparameter optimization. In *Proceedings of Advances in Neural Information Processing Systems*.

Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. *arXiv preprint arXiv:1909.02768*.

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 Empirical Methods in Natural Language Processing*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. *Proceedings of the Association for Computational Linguistics*.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. XGLUE: A new benchmark dataset for cross-lingual pretraining, understanding and generation. *arXiv preprint arXiv:2004.01401*.

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. In *Proceedings of the 57th Annual Meeting of the 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 European Chapter of the Association for Computational Linguistics*.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of Empirical Methods in Natural Language Processing*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Yevgeni Berzak, Riyaz Ahmad Bhat, Eckhard Bick, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Claudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Jessica Kenney, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lucia Lam, Phuong Lê Hồng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Keiko Sophie Mori, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lu'o'ng Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Petya Osenova, Robert Östling, Lilja Øvrelid, Valeria Paiva, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Baiba Saulīte, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Carolyn Spadine, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Mats Wirén, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2016. Universal Dependencies 1.4.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. *Proceedings of Empirical Methods in Natural Language Processing*.

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 Association for Computational Linguistics*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of Empirical Methods in Natural Language Processing*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the Association for Computational Linguistics*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for ner. In *Proceedings of the Association for Computational Linguistics*.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Proceedings of Advances in Neural Information Processing Systems*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of International Conference on Computational Linguistics*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the Association for Computational Linguistics*.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. In *Linguistic Data Consortium*. LDC2006T06.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*.

Mengxzhou Xia, Guoqing Zheng, Subhabrata Mukherjee, Milad Shokouhi, Graham Neubig, and Ahmed Hassan Awadallah. 2021. MetaXL: Meta representation transformation for low-resource cross-lingual learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the Association for Computational Linguistics*.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime G Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of Empirical Methods in Natural Language Processing*.

Wei Ye, Bo Li, Rui Xie, Zhonghao Sheng, Long Chen, and Shikun Zhang. 2019. Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data. In *Proceedings of the Association for Computational Linguistics*.

A Appendix

A.1 Data for Relation Extraction and Argument Role Labeling

In this section, we describe details of the dataset for RE and ARL. Table 7 reports the statistics of the dataset and Table 8 summarizes references and baseline results.

We create a dataset using the ACE2005 corpus (Walker et al., 2006), which more closely replicates the setting a model will be faced with in a real-world information extraction scenario. First, we shuffle documents into 80%/10%/10% splits for train/dev/test, then extract candidate entity-pairs from each document. For RE, the first approach in Ye et al. (2019) is adopted to extract negative instances. Negative instances whose entity-type combination has never appeared as a positive example in the training data are filtered out. For ARL, we create negative instances by pairing each trigger with every entity in a sentence. Details on the two datasets are summarized in Table 7.

As a baseline for the dataset, we reimplement the Graph Convolutional Network (GCN) model of Subburathinam et al. (2019) using multilingual embeddings learned by fastText (Bojanowski et al., 2017) on Wikipedia (GCN_{ReImp}). Tables 8 display F1 for zero-shot cross-lingual transfer.

Task	Lang	Train	Dev	Test	Pos/Neg
RE	en	63177	10218	6861	1:8.9
	zh	57824	7096	8162	1:9.4
	ar	32984	4482	4638	1:8.9
ARL	en	21875	3345	2603	1:2.6
	zh	15095	2226	2017	1:2.7
	ar	11587	1221	1568	1:2.9

Table 7: Statistics of the dataset. Number of instances and the total positive/negative ratio.

A.2 Variance of Different Meta-train/dev/test Split is Relatively Low

In this section, we present a statistical analysis of model selection results for POS and QA between different meta-train/dev/test splits. Table 9 shows LMS on average improves a point of 0.74 relative to En-Dev and a point of 0.44 relative to Pivot-Dev. We found that the variance in end-task performance between different meta-train/dev/test splits is relatively low.

	RE (F_1)		ARL (F_1)	
	ar	zh	ar	zh
GCN_{ReImp}	39.43	32.74	16.48	23.49
Model Selection				
En-Dev	36.10	67.68	44.11	60.96
LMS	**39.54**	**70.75**	**47.08**	**62.05**
100-Target	34.68	68.20	44.11	62.52
All-Target	41.92	69.13	47.15	63.81

Table 8: F1 scores for relation extraction and argument role labeling on the test set. En-Dev/100-Target/All-Target: model selection based on the highest F1 of English dev set/100 target language dev set examples/target language dev set. Ours: model selection based on the highest scores for the target language: $\arg\max_m s(m, l_{target})$.

We train a single LMS with pivot languages in $\{ar, de, es, nl, zh\}$ for POS and $\{ar, de, es, zh\}$ for QA, and test it on all the target languages. All the results are reported with mean and standard deviation with five runs (different meta-train/dev/test splits). A Z-test is performed to the differences between LMS/Pivot-Dev and En-Dev. LMS is statistically significantly (p-value ≤ 0.05) higher than En-Dev baseline across all languages and two tasks while Pivot-Dev fails in three languages. LMS also obtains a lower standard deviation for the model scores except for Swedish (sv) and Vietnamese (vi).

A.3 Does this Approach Generalize to XLM-RoBERTa?

In Section 6, we showed that our approach consistently selects better fine-tuned models than those chosen using English development data. To test the robustness of our approach with a different multilingual pre-trained Transformer, we re-train and evaluate using XLM-RoBERTa-base (Conneau et al., 2019), with the same settings used for mBERT in Section 6 for RE and ARL.

RE In the left section of Table 10, our approach selects a model with a higher F1 score compared to En-Dev in Chinese and on par with En-Dev in Arabic.

ARL In the right section of Table 10, our approach selects a model with a higher F1 score compared to En-Dev in Arabic but performs worse on Chinese (En-Dev outperforms the All-Target). Overall, our approach appears to be effective when used with XLM-RoBERTa.

Task	Lang	Ref	En-Dev	Pivot-Dev	LMS	100-Target	All-Target	# All-Target
POS (Acc)	bg	87.4	$87.22_{\pm 0.24}$	$\underline{87.35}_{\pm 0.37}$ (es)	$\mathbf{87.75}_{\pm 0.14}$*	$87.72_{\pm 0.50}$	$88.09_{\pm 0.17}$	1115
	da	88.3	$88.59_{\pm 0.17}$	$\mathbf{88.81}_{\pm 0.12}$ (nl)*	$\underline{88.74}_{\pm 0.14}$*	$88.74_{\pm 0.06}$	$89.01_{\pm 0.18}$	322
	fa	72.8	$\underline{71.58}_{\pm 0.28}$	$71.55_{\pm 1.02}$ (es)	$\mathbf{73.57}_{\pm 0.13}$*	$73.00_{\pm 0.76}$	$73.80_{\pm 0.31}$	599
	hu	83.2	$\underline{82.81}_{\pm 0.37}$	$82.34_{\pm 0.27}$ (de)	$\mathbf{83.28}_{\pm 0.17}$*	$83.14_{\pm 0.17}$	$83.18_{\pm 0.15}$	179
	it	84.7	$84.44_{\pm 0.45}$	$\underline{84.68}_{\pm 0.48}$ (es)*	$\mathbf{85.04}_{\pm 0.19}$*	$85.14_{\pm 0.27}$	$85.61_{\pm 0.21}$	489
	pt	82.1	$81.87_{\pm 0.16}$	$\underline{82.06}_{\pm 0.23}$ (es)*	$\mathbf{82.18}_{\pm 0.08}$*	$82.14_{\pm 0.34}$	$82.36_{\pm 0.17}$	271
	ro	84.7	$83.84_{\pm 0.30}$	$\mathbf{85.59}_{\pm 0.48}$ (es)*	$\underline{84.74}_{\pm 0.04}$*	$84.79_{\pm 0.47}$	$85.39_{\pm 0.16}$	1191
	sk	83.6	$83.49_{\pm 0.42}$	$\mathbf{84.09}_{\pm 0.50}$ (es)*	$\underline{83.93}_{\pm 0.28}$*	$84.07_{\pm 0.58}$	$84.90_{\pm 0.33}$	1060
	sl	84.2	$\underline{84.26}_{\pm 0.34}$	$84.10_{\pm 0.57}$ (es)	$\mathbf{84.91}_{\pm 0.24}$*	$84.23_{\pm 0.56}$	$85.41_{\pm 0.61}$	735
	sv	91.3	$91.37_{\pm 0.05}$	$\mathbf{91.63}_{\pm 0.19}$ (nl)*	$\underline{91.55}_{\pm 0.21}$*	$91.58_{\pm 0.23}$	$91.73_{\pm 0.05}$	504
QA (F_1)	hi	43.8	$39.93_{\pm 1.45}$	$41.40_{\pm 1.35}$(es)*	$42.09_{\pm 0.91}$*	$40.04_{\pm 0.81}$	$42.56_{\pm 0.26}$	507
	vi	57.1	$57.18_{\pm 0.82}$	$57.66_{\pm 2.03}$(ar)	$57.73_{\pm 1.16}$*	$58.83_{\pm 0.90}$	$59.12_{\pm 0.92}$	511

Table 9: Model scores (mean $\pm$ sd) selected based on LMS for POS and QA over 5 runs. Bold indicates the best score and underline indicates the second best. ∗ indicates the LMS/Pivot-Dev is statistically significantly (p-value ≤ 0.05) higher than En-Dev.

	RE (F_1)		ARL (F_1)	
	ar	zh	ar	zh
GCN$_{\text{ReImp}}$	39.43	32.74	16.48	23.49
mBERT	36.10	67.68	44.11	60.96
Model Selection				
En-Dev	**40.79**	64.48	50.65	**62.73**
LMS	**40.79**	65.11	**52.96**	61.92
100-Target	42.33	65.38	52.90	62.12
All-Target	44.66	65.75	53.09	62.27

Table 10: XLM-RoBERTa experiment: F1 of relation extraction and argument role labeling. Model selection results are based on XLM-RoBERTa-base models in the meta-test set.

A.4 Can Multi-task Learning Help?

Our setting does not assume access to the labeled data in the target language for a particular task. However, labeled data in the target language may be available for a relevant auxiliary task, which could help the scoring function learn to better estimate whether a model is a good match for the target language.

To test whether an auxiliary task in the target language might help to select a better model for the target task, we train the LMS on two tasks: RE and ARL. Gold rankings on the models are then computed for each language using the pivot languages' dev sets. Also, another "silver" ranking is computed for each language using the auxiliary task. The scoring function is then trained to rank mBERT models for both tasks. To differentiate the two tasks, a variant of the scoring function, $s(m, l, t)$, which concatenates a randomly initialized task embedding with the language embedding is adopted.

In Table 11, our approach selects a model with a higher F1 score for RE. However, multi-task does not benefit ARL but still outperforms En-Dev. As for future direction, we believe an LMS that is trained on an auxiliary dataset can be transferred to the target dataset, hence release the requirement of a small amount of pivot language development data in the target dataset.

Task	*En-Dev*	([Pivot], lang2vec)	+ Multi-task
RE	51.81	55.92	**57.31**
ARL	50.98	**53.60**	51.99

Table 11: Multi-task analysis using additional training data in the target language from another task. ([Pivot], lang2vec): baseline of training within a single task data. Model selection is based on the highest scores for the target language and target task: $\arg\max_m s(m, l_{\text{target}}, t_{\text{target}})$