How to Translate Your Samples and Choose Your Shots? Analyzing Translate-train & Few-shot Cross-lingual Transfer

Anonymous ACL submission

Abstract

Translate-train or few-shot cross-lingual transfer can be used to improve the zero-shot performance of multilingual pretrained language models. Few-shot utilizes high-quality low-quantity samples (often manually translated from the English corpus). Translate-train employs a machine translation of the English corpus, resulting in samples with lower quality that could be scaled to high quantity. Given the lower cost and higher availability of machine translation compared to manual professional translation, it is important to systematically compare few-shot and translate-train, understand when each has an advantage, and investigate how to choose the shots to translate in order to increase the few-shot gain. This work aims to fill this gap: we compare and quantify the performance gain of few-shot vs. translate-train using three different base models and a varying number of samples for three tasks/datasets (XNLI, PAWS-X, XQuAD) spanning 17 languages. We show that scaling up the training data using machine translation gives a larger gain compared to using the small-scale (higher-quality) few-shot data. When few-shot is beneficial, we show that there are random sets of samples that perform better across languages and that the performance on English and on the machine-translation of the samples can both be used to choose the shots to manually translate for an increased few-shot gain.[1]

1 Introduction

With the emergence of large-scale multilingual Pretrained Language Models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), a significant amount of research went into exploring the cross-lingual transfer capabilities of these models, allowing for an easier adaptation to a task in many various languages. This is achieved through a number of approaches.

Zero-shot cross-lingual transfer has become a research focus, e.g. XTREME / XTREME-R benchmark (Hu et al., 2020; Ruder et al., 2021). In this approach, transfer to new languages is done by fine-tuning a multilingual PLM on the task at issue, using only an English corpus (source language) and reporting the performance on multiple target languages.

Few-shot cross-lingual transfer was recently shown to give an advantage over zero-shot cross-lingual transfer (Lauscher et al., 2020). In this approach, it's shown that fine-tuning the model using a small amount of target-language task data (few-shot) improves the performance, especially for low-resource languages.

Translate-train is another common approach to improve the performance. Here the full training dataset is machine translated to the target language and used for fine-tuning. There exists relatively good Machine Translation (MT) systems for the languages that are usually studied in the few-shot approach[2] that could be used in translate-train.

In the following, we use few-shot to refer to fine-tuning using fewer samples of high-quality professional manual translation. Translate-train is used to refer to fine-tuning using lower-quality machine translation that has the potential to be scaled to a larger number of samples. Although some research has dealt with few-shot cross-lingual transfer and analyzing it (Lauscher et al., 2020; Zhao et al., 2021), no systematic study was done to compare it to translate-train. Given that both zero-shot and few-shot cross-lingual transfer assume the availability of a large-scale English corpus of the task for source training, we hypothesize that the *translate-train approach might have an advantage over few-shot given the scale of data that would be available even if not at the best quality.*

[1]Our code will be published under: https://www.gihtub.com/***

[2]All target languages in the studied datasets are supported by e.g. Google Translate: https://cloud.google.com/translate/docs/languages

On the other hand, When there is a need for few-shot cross-lingual transfer for some task and therefore a need for professional translation of some training samples, this entails significantly more effort and cost compared to MT. It is then important to *find out which samples to manually translate given the high variance in performance depending on the choice of samples* (Zhao et al., 2021).

We investigate both those research directions using 3 base models (mBERT$_{base}$, XLM-R$_{base}$, XLM-R$_{large}$) on 3 high-level semantic tasks and datasets, XNLI (Natural Language Inference), X-PAWS (Paraphrase Detection) and XQUAD (Question Answering), spanning 17 diverse languages. We investigate the following research questions:

Q1. How does the performance of few-shot cross-lingual transfer compare to that of translate-train? We show that there is a performance advantage for few-shot transfer over translate-train given the same number of samples, but that with the increase of samples used for translate-train, this gap shrinks, and using the full large-scale corpus in translate-train results in a clear advantage over few-shot. We show that at a scale of 10x-100x of machine-translation to manual-translation, quantity trumps quality and it's recommended in this case to use translate-train if MT is available for the language. Few-shot transfer still has an advantage when less source data is available and it's therefore not possible to benefit from the scale gain of using MT.

Q2. Are there sets of samples that have better few-shot performance if translated and how can those sets be identified? We show that when few-shot transfer is beneficial for the task, there are random sets of samples that perform better across most target languages and across different model initializations. We investigate using the performance on the English version of the samples and the machine-translated version to choose the best candidates to manually translate and use for few-shot transfer. We show that there is a correlation between the performance of the same set of shots across languages and that the few-shot samples that perform well on the source language, English, usually perform also better across languages and on average. The same correlation is seen between the performance of manual and machine translation. We show empirically that choosing the sets of samples for few-shot transfer using those heuristics or a model based on those features results in more bang for your shots.

2 Related Work

Cross-lingual transfer: The cross-lingual transfer capabilities of multilingual pretrained language models have led to major recent advances and a growing number of such models have been introduced, e.g., mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), mT5 (Xue et al., 2021) etc. The cross-lingual transfer is usually exploited in a zero-shot setup, and benchmarks are built based on this assumption e.g. XTREME/XTREME-R (Hu et al., 2020; Ruder et al., 2021).

Few-shot: There has been recently some focus on few-shot cross-lingual transfer and its analysis. Lauscher et al. (2020) shows the effectiveness of few-shot compared to zero-shot cross-lingual transfer especially in lower-resource and distant languages, where zero-shot is least effective and few-shot gives a large gain. Zhao et al. (2021) analyzes few-shot cross-lingual transfer emphasizing that the choice of shots has a significant effect on the performance. The experiments are conducted at a small scale of around 10 samples. Compared to this, we conduct larger-scale few-shot experiments with a size up to hundreds of samples and focus on choosing the best-performing samples.

Translate-train: is commonly used to boost the performance for a target language using a machine translation of the source corpus (Conneau et al., 2018; Lample and Conneau, 2019; Conneau et al., 2020; Hu et al., 2020). Xue et al. (2021) shows that, similar to zero-shot, translate-train performance increases with the scale of the model. No systematic study tested the effect of the scale of the translated data in comparison with few-shot to understand the interplay of data quality vs. quantity in this context.

Choosing samples: Two related areas are sample selection (Rousseeuw, 1984) which is used for robust training on noisy data (Song et al., 2019) and active learning (Cohn et al., 1994; Krogh and Vedelsby, 1994) used to choose the best potential samples to annotate (Siddhant and Lipton, 2018). Both assume access to the actual sample input (with or without label). On the other hand, this work investigates choosing samples while only having access to the source-language sample input/output.

3 Datasets

We focus on high-level tasks and conduct our experiments on 2 classification tasks and a question answering task (Table 1) from the XTREME benchmark. The details and properties of the languages

Dataset	|Train|	|mDev|	|mTest|	|Langs|	metric
XNLI	392,702	2,490	5,010	15	Acc
PAWS-X	49,401	2,000	2,000	7	Acc
XQuAD	87,599	261	1,190-261= 930	11	F1

Table 1: Datasets statistics. Train is the English training dataset. |mDev| and |mTest| are used to indicate the size of the multilingual split of the dataset.

can be found in Appendix Table 6. When attempting to choose the shots, we rely on measuring the performance of the same set of samples across different languages, so we are limited to datasets with parallel corpus, i.e. the target language corpus is created by translating the English corpus as opposed to collecting and annotating the target language corpus from scratch: **XNLI** (Conneau et al., 2018) is a professional translation of the dev and test set of the MultiNLI dataset (Williams et al., 2018) into 14 languages. The dataset consists of pairs of sentences, a premise and a hypothesis, where the task is to predict whether the premise entails, contradicts, or is neutral to the hypothesis. **PAWS-X** (Yang et al., 2019) dataset is a professional translation of the dev and test set of the PAWS dataset (Zhang et al., 2019) into 6 languages. The dataset consists of pairs of sentences and the task is to predict whether those two sentences are paraphrases of each other. **XQuAD** (Artetxe et al., 2020b) is a professional translation of the dev set from SQuAD v1.1 (Rajpurkar et al., 2016) into 10 other languages. The dataset consists of a paragraph and a set of questions. The task is to select the span of the paragraph that answers the question. 10 paragraphs from the multilingual split are reserved, similar to Lauscher et al. (2020), as dev (total of 261 question/answer samples) and the rest as test set.

4 Experiments

Three main models are used: mBERT (base), XLM-R_B (base) and XLM-R_L (large). We report results on XLM-R_B if not specified otherwise, because it strikes a balance between good performance and efficient training. For each task, we fine-tune the model on the source language (English) corpus for 5 epochs with early stopping using the loss on the English dev set. We then continue fine-tuning the model on the target language either in a few-shot or translate-train setup as explained in the following sections. Training details are in Appendix A.

4.1 Few-shot experiments

We use samples from the multilingual dev set as training samples. Few-shot fine-tuning is done as follows: for each language, we separately continue fine-tuning the source model for one epoch on $n \in \{10, 50, 100, 500, 1k\}$ samples from the target language corpus for the two classification tasks and for $n \in \{10, 50, 100, 250\}$ for the Question Answering task, given the smaller amount of data available for training in this case. We report the results on the test set for each target language. For each n number of samples, the performance is averaged across 5 different sets of random samples using 5 different fine-tuned models with different random initializations, 25 runs in total. This is to ensure more robust results when measuring the gain over zero-shot given the high variance across different sets of samples (Zhao et al., 2021) as well as the variance in zero-shot performance across all random initializations (Keung et al., 2020). For comparing the performance across shots, we make sure to use the same set of parallel samples across languages, using the sample ids, to compare how a set of samples performs when translated to different languages. This is possible due to our selection of tasks and datasets that have a parallel corpus for the various target languages.

4.2 Translate-train experiments

We train using MT of the source train set to each target language[3] and adapt a similar setup as few-shot: for each language, continue fine-tuning separately on $n \in \{10, 50, 100, 500, 1k, 10k, |dataset|\}$ samples from the machine-translated train set and report the results on the test set of the target language.

5 Results

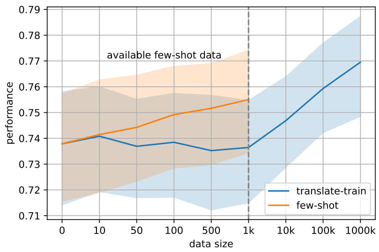
5.1 How to translate your samples? Few-shot vs. translate-train

To demonstrate the full potential for each approach, Table 2 shows a **performance summary** for zero-shot, few-shot and translate-train when the maximum possible number of samples is used. The gap to English performance is the average of the gap between the target language performance and the performance on the English test set. Both few-shot and translate-train help bridge the gap, but using translate-train on a large scale has an advantage in

[3]We use the Machine Translation provided by the XTREME Benchmark: https://console.cloud.google.com/storage/browser/xtreme_translations

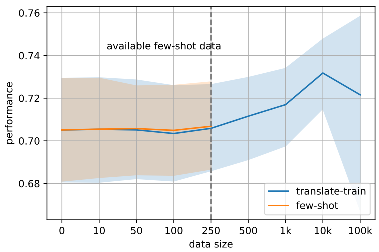
	XNLI			PAWS-X			XQuAD		
	langs avg	en gap	gain	langs avg	en gap	gain	langs avg	en gap	gain
				English performance					
mBERT	81.85±0.99	-	-	92.81±0.33	-	-	83.74±0.36	-	-
XLM-R$_B$	84.04±0.65	-	-	93.99±0.35	-	-	83.10±0.29	-	
XLM-R$_L$	88.98±0.29	-	-	95.13±0.35	-	-	87.07±0.59	-	-
				zero-shot (only en training)					
mBERT	65.75±0.31	-16.10	-	81.24±1.58	-11.57	-	61.51±0.24	-22.22	-
XLM-R$_B$	73.79±0.34	-10.26	-	82.08±0.92	-11.92	-	70.51±0.32	-12.60	-
XLM-R$_L$	79.61±0.61	-9.37	-	85.89±0.59	-9.24	-	76.98±0.25	-10.10	-
				few-shot (max 1k high-quality translated samples)					
mBERT	67.96±0.29	-13.89	2.21	83.01±0.91	-9.80	1.76	65.06±0.27	-18.67	3.55
XLM-R$_B$	75.50±0.30	-8.54	1.71	82.83±0.53	-11.16	0.76	70.68±0.28	-12.42	0.17
XLM-R$_L$	81.70±0.14	-7.27	2.10	86.73±0.17	-8.40	0.84	77.06±0.16	-10.01	0.08
				translate-train (full machine-translated training set)					
mBERT	72.81±0.24	-9.04	7.06	85.74±0.67	-7.07	4.50	69.84±0.34	-13.89	8.33
XLM-R$_B$	76.95±0.32	-7.09	3.16	85.06±0.66	-8.93	2.99	72.16±0.16	-10.95	1.65
XLM-R$_L$	**82.46±0.19**	-6.51	2.86	**88.73±0.20**	-6.40	2.83	**77.26±0.35**	-9.81	0.29

Table 2: **Performance summary**. The average performance on all languages along with the gap to the English performance and the gain over zero-shot performance for the same model



(a) Average Accuracy on XNLI across languages

(b) Average Accuracy on PAWS-X across languages



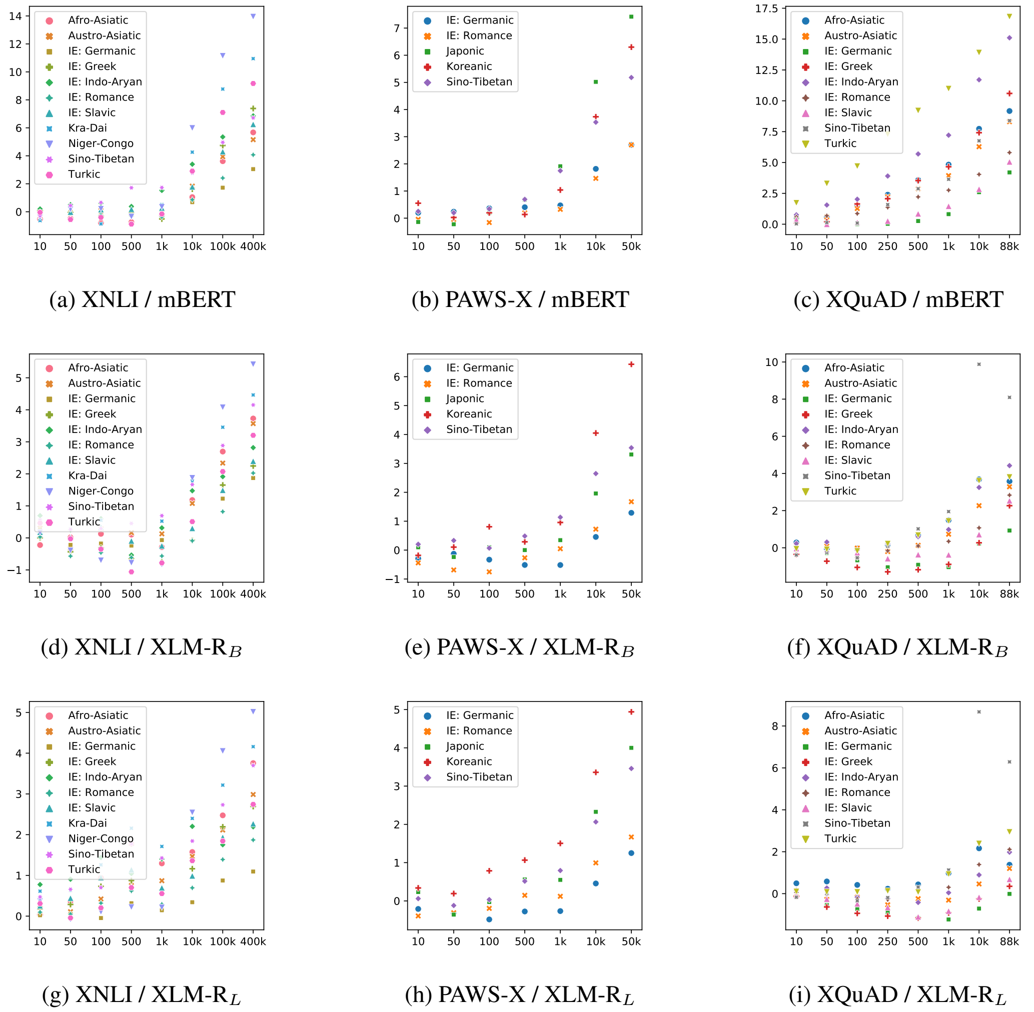
(c) Average F1 on XQuAD across languages

Figure 1: Avg performance across langs for **translate-train vs. few-shot** using **XLM-R$_B$**. The biggest performance boost comes from using translate-train

further narrowing the gap as compared to the small scale of few-shot transfer. This results in translate-train having the best performance for all models across all datasets. The highest *gain* is seen for the model with the highest *en gap* (mBERT) for both few-shot and translate-train. For XLM-R on XQuAD, the gain is low and negligible. Given that there is a significant gain for mBERT and the same experimental setup is used for all models, the lack of gain is probably not dataset-specific and could happen with some models.

To see the effect of the available dataset size in each scenario, Figure 1 shows the average performance across languages for **few-shot vs. translate-train** across varying number of samples. We can see an advantage of having manual over machine translation resulting in a clear performance gap between both on XNLI for the same number of samples. This gap increases with the increase of the number of samples as seen at 1k. The availability of manual translation for few-shot is limited though and starting from 10k-100k, the scale of translate-train has an advantage for all tasks (similar results for the other models are in Appendix Figure 6, 7). The performance on PAWS-X and XQuAD does not improve much with few-shot as shown in Figure 1b and 1c, and the clear boost comes from using the large scale machine-translated dataset. We discuss the observed large variance on XQuAD across languages in the following.

Detailed results & language analysis Figure 2 shows the translate-train performance gain (over zero-shot) across a varying sample size for each **language family** (More details about the languages in Appendix Table 6). We can see, across all tasks and models, that European languages have a small gain compared to non-European languages which show the largest gain e.g. Swahili (Niger-kongo) in XNLI, Korean and Japanese in PAWS-X, and Turkish and Chinese for XQuAD. Those languages also tend to have a larger zero-shot performance gap to English and are more distant to it (the source language). Those results are comparable to the few-shot results of Lauscher et al. (2020). We can see that the languages with the most gain differ between mBERT and XLM-R mainly because XLM-R extends the pre-training corpus using CommonCrawl to have more data that less-spoken languages benefit especially from e.g. Turkish zero-shot performance on XQuAD is low with mBERT as compared to XLM-R models which result in more gain for Turkish with mBERT on XQuAD. (detailed results on XQuAD in Appendix Figure 15, 14, 16). Appendix C contains the detailed perfor-



(a) XNLI / mBERT (b) PAWS-X / mBERT (c) XQuAD / mBERT

(d) XNLI / XLM-R_B (e) PAWS-X / XLM-R_B (f) XQuAD / XLM-R_B

(g) XNLI / XLM-R_L (h) PAWS-X / XLM-R_L (i) XQuAD / XLM-R_L

Figure 2: **Translate-train gain on language families.** European languages especially Germanic have the least gain, whereas various other families like Koreanic and Niger-Kongo have the most gain (detailed results in Appendix C)

mance gains for few-shot and translate-train over zero-shot for each language across varying sizes of samples. We can see from those results that once the full machine-translated training set is used, a clear advantage for translate-train is seen across almost all languages and in all tasks. We can see that the gain for Urdu (ur) is the highest on XNLI up until 100k when it starts decreasing. We think this might be due to a lower-quality MT. The same effect is seen for Thai (th) on XQuAD with a significant performance degrade when the full training dataset is used (details in the Appendix in Figure 15). This is also the reason for the degrade and high variance seen at this point in Figure 19b.

We investigate whether longer training would have changed the results and would have been beneficial, especially for few-shot where longer training on the high-quality manual translation might be beneficial. We split the available set of samples into train/dev and train for 10 epochs with early stopping on dev. Although some languages benefit from this setup, it still yields comparable results and translate-train still has a clear advantage. (results in Appendix Figure 17 and 18).

5.2 How to choose your shots? Which samples to translate for few-shot?

Few-shot can still have an advantage over translate-train when the English dataset is not large enough to benefit from the scale effect of translate-train. It can also be necessary when adapting to a target language that does not have an existing machine translation system or does not have a good one. Creating few-shot samples, in this case, can be done by collecting and labeling new samples or by translating samples from the available English dataset. The latter is a common method and 4 out of the 7 non-retrieval datasets in XTREME use manual professional translation to create samples in the target languages (all of which high-level semantic tasks). It is beneficial then to support in selecting the samples with higher-performance potential to translate and do few-shot training on.

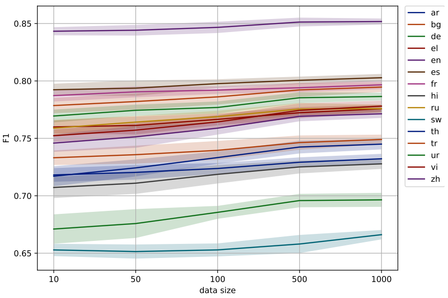


Figure 3: **XNLI accuracy variance on different shots**. High variance decreases with an increased data size

To emphasize the significance of choosing the samples, we plot in Figure 3 the XNLI **performance variance on different shots** (using the same model initialization) across 20 sets of random few-shot samples varying in size from 10 to 1k samples. The performance varies, sometimes significantly, depending on the set of samples used. Zhao et al. (2021) shows similar variance observations on a smaller number of samples (around 10). We consider a larger size range that is more representative of the data size if a manual translation is conducted. The performance variance across shots decreases with the increased number of shots. This means that choosing the shots to translate is more important when smaller size of samples is used. (similar results on PAWS-X and XQuAD are in Appendix Figure 19 although for XQuAD the variance increases with the size). In the following, we focus mainly on XNLI as the task that had the most few-shot gain. We investigate whether there are sets of samples that have a potential for better performance across languages and what could be an indication of that. For a set of shots, we consider two indicators: the performance of this set in another language, and the performance on the MT of the samples in the set.

5.2.1 Correlation between performance across languages

If the performance of a set of samples for one language can be an indication of its performance on another language, a high correlation between the performance for both languages is expected. To estimate this, we calculate the performance using the manual translations across languages of the same set of training samples. We then calculate the Pearson correlation of the performance across 5 random sets of samples (with varying sample-set sizes) using 5 models with different random initialization. As seen in Figure 4, there is a high positive correlation between the performance on XNLI for the various languages (using XLM-R$_B$). This is also the case, but to a lesser degree for PAWS-X as seen in the Appendix Table 8. XQuAD, on the other hand, has low and sometimes even negative correlation (Appendix Table 11), which might be due to the nature of the task or the fact that we have less data in this case for both training and test. It is also worth noting that the correlation is lower for both tasks, PAWS-X and XQuAD, which had low few-shot gain.

A breakdown of the English correlation based on data size is show in the Appendix Table 7 and 9. As an example of this, Figure 5 shows **XNLI few-shot gain** over zero-shot performance for 5 random sets of samples $\{A, B, C, D, E\}$ each containing 10 samples. The performance is shown for 3 different XLM-R$_B$ initializations. The sets $\{A, C, E\}$ perform better than $\{B, D\}$ across target languages and on average as well as across different initializations. The performance on English can be used as an indicator of the best shots to choose as seen when comparing the English performance (top) to the average (bottom, excluding the English performance). This is here the case even when further fine-tuning a model on English samples results in a decreased English performance as seen for the 2nd model initialization. The least negative sets of samples still correspond to the best performing shots. The results generalize for varying sizes of few-shot sets e.g. 1000 samples in Appendix Figure 20.

5.2.2 Correlation between manual and machine translation performance

Another possible indicator of the best performing set of samples could be the performance of the samples in the set when they are machine translated to the target language. Artetxe et al. (2020a) has shown that subtle patterns in the (machine or manual) translated samples can have a notable impact on the model performance, so it is important to empirically study the relation between both. Similar to the above, we calculate the correlation between the performance for both manual and machine translation of the same set of samples for each target

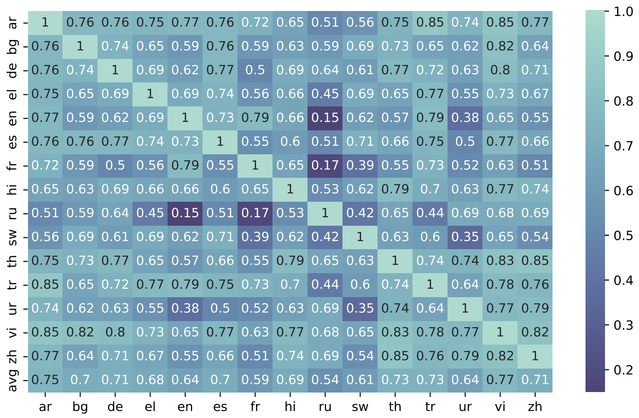


Figure 4: **XNLI Pearson correlation** between the performance on **English** and the performance on other languages.

ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh
0.65	0.86	0.89	0.75	0.88	0.88	0.80	0.75	0.72	0.85	0.86	0.61	0.89	0.89

Table 3: **XNLI Pearson correlation** between the performance of **machine translation** and manual translation

Figure 5: **XNLI few-shot gain** over zero-shot across 5 sets of samples (size=10) for 3 different model initializations. Sets C, A, and E yield better performance for the 3 different initializations. The English performance can be used as an indicator when choosing samples to translate

language. As seen for XNLI in Table 3, there is an even higher correlation than with the English performance. A somewhat lower correlation is seen for PAWS-X in Appendix Table 10. Lower correlation might be a result of lower-quality MT or a result of the different patterns introduced by MT as mentioned before.

5.2.3 Gain from choosing shots

We show in Table 4 the **few-shot performance gain** resulting from choosing the shots with the highest English performance and the highest MT performance. Random samples are used for few-shot cross-lingual transfer in related work, so we compare to the average few-shot gain across the different shots in *no choosing (avg)*, and also to the minimum in *no choosing (min)*, because an important aspect of choosing shots is avoiding the worst-performing ones (Comparing to the average hides the fact that we might accidentally use a very bad set of shots). We can see a clear gain in most cases across all models when using *en performance* or *mt performance*. When there is no gain compared to *no choosing (avg)*, the performance is still comparable and the benefit of not choosing the worst performing shots is still there as compared to *no choosing (min)*. The few-shot gain with chosen-shots is most significant at smaller number of samples where the gain is almost double that from *no choosing (avg)*.

Combining both En and MT performance when choosing the shots is expected to result in more gain, so we investigate feeding the performance values as features to a **linear model** that takes as input the performance of a set of samples and predicts the performance gain when this set is manually translated and used for few-shot. Predicting the performance gain is also helpful to avoid translating any set of samples if all are expected to result in a negative or low gain. We use the performance metrics as a dataset: collecting the performance of En/MT of random sets of samples along with the performance of the actual manual translation. This is done using 5 different random sets of samples for 5 different XLM-R$_B$ initialization with varying sample sizes across all languages (excluding English) resulting in 1750, 750, 1100 data points for XNLI, PAWS-X and XQuAD. For each language, we train the model using the data from all other languages and evaluate on the selected language. Cross-validation is done on the data after excluding the selected language to choose the best hyperparameters. The following **features** are considered as input: En and/or MT performance gain for the set of samples corresponding to each data point. In all cases, we consider: the zero-shot performance (since the gain is usually larger when the zero-shot performance is lower), and the number of samples used for that data point. We also investigate whether adding **language features**[4] can improve the prediction in a way similar to the analysis by Lauscher et al. (2020). lang2vec [5] from Littell et al. (2017) is used to obtain the feature vectors for each language. The cosine similarity between the English vectors and the vectors for each language are added as 5 new scalar features (values are in Appendix Table 6). Those features can help the model better use the English performance depending on the similarity between the language and English. The prediction error of the linear models is reported in Appendix Table 13.

We can see in Table 4 that using the models improves the chosen-shots performance gain for XNLI with the best result, as before, using a combination of all features. This is not the case for PAWS-X and could be partially due to having a smaller performance data and fewer languages to train on (7 as compared to 15 languages for XNLI). The detailed results for the different languages are in the Appendix Figure 21. Choosing the shots improves the few-shot performance on XNLI for all languages across almost all sample sizes. For PAWS-X, there is mixed gain/loss but the improvement when using English performance at maximum size is concentrated in the European languages.

[4]syntax, phonology, inventory, family and geographical location as features

[5]https://github.com/antonisa/lang2vec

		10	50	100	$500^{(*)}$	1000
			mBERT			
XNLI	no choosing (avg)	0.22	0.51	0.92	1.35	2.21
	no choosing (min)	-0.18	-0.00	0.17	0.11	0.99
	en performance	0.59	0.84	1.06	1.37	2.43
	mt performance	**0.76**	**0.87**	**1.23**	**1.87**	**2.52**
PAWS-X	no choosing (avg)	0.22	-0.02	-0.02	0.89	1.76
	no choosing (min)	0.04	-0.24	-0.28	0.24	0.67
	en performance	0.03	0.37	-0.02	1.21	1.89
	mt performance	**0.46**	**0.54**	**0.27**	**0.72**	**2.00**
XQuAD	no choosing (avg)	0.64	**1.68**	2.28	**3.55**	
	no choosing (min)	-0.15	-0.35	-0.59	-0.59	
	en performance	**0.80**	1.58	**2.34**	3.52	
			XLM-R$_B$			
XNLI	no choosing (avg)	0.36	0.64	1.13	1.38	1.71
	no choosing (min)	0.04	-0.15	0.10	0.36	0.65
	en performance	0.71	1.15	1.32	1.82	1.90
	mt performance	**0.88**	1.08	1.36	1.81	2.01
	en + mt model	0.85	1.11	1.42	**1.85**	2.01
	+ lang features	0.83	**1.13**	**1.44**	**1.85**	**2.03**
PAWS-X	no choosing (avg)	0.19	0.02	-0.20	0.44	0.76
	no choosing (min)	-0.34	-0.43	-1.05	-0.23	0.10
	en performance	0.17	0.10	**0.23**	**0.53**	0.71
	mt performance	**0.38**	**0.19**	0.09	0.42	0.73
	en + mt model	0.32	0.09	0.13	0.44	0.76
	+ lang features	0.26	0.04	0.00	0.52	**0.84**
XQuAD	no choosing (avg)	**0.04**	0.06	**-0.02**	**0.17**	
	no choosing (min)	-0.36	-0.71	-0.76	-1.31	
	en performance	**0.04**	**0.08**	-0.15	**0.17**	
			XLM-R$_L$			
XNLI	no choosing (avg)	0.53	0.76	1.35	**1.84**	2.10
	no choosing (min)	0.28	0.32	0.70	0.97	1.29
	en performance	**0.71**	**1.09**	**1.51**	1.78	**2.11**
PAWS-X	no choosing (avg)	**0.06**	-0.10	-0.52	0.46	0.84
	no choosing (min)	-0.56	-0.75	-0.78	0.02	0.04
	en performance	-0.14	**0.27**	**0.18**	**0.55**	**1.04**
XQuAD	no choosing (avg)	0.05	0.06	-0.08	**0.08**	
	no choosing (min)	-0.20	-0.42	-0.97	-1.17	
	en performance	**0.09**	**0.10**	-0.01	**0.06**	

(*) 250 for XQuAD

Table 4: **Chosen-shots performance gain**. Gain over zero-shot performance when choosing the best set of shots using a heuristic (en or mt performance) or a linear model that predicts the performance.

6 Conclusion and Future Work

This work conducted a systematic comparison between translate-train and few-shot cross-lingual transfer. It quantified the performance gain for each and showed that starting from 1k samples, machine-translated data could be used to improve over zero-shot performance, and that at 10k-100k, there's an advantage for translate-train over few-shot. For the tasks that benefit from few-shot, we show that there are random sets of samples that perform better across languages and that the English performance of the samples in those sets can help us identify them. The performance of the MT of the samples can also be used as another indicator. When not incurring gain, both help at least avoid the worst performing samples.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020a. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020b. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1994. Active learning with statistical models. In *NIPS*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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.

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.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554, Online. Association for Computational Linguistics.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation, and active learning. In *NIPS*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

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.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Peter Rousseeuw. 1984. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880.

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.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 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 the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. 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 5751–5767, Online. Association for Computational Linguistics.

A Training Details

Software: We use the Huggingface Transformers [6] for fine-tuning the pretrained language models. We use scikit-learn [7] to train the performance prediction models. Our code will be made publicly available [8].
Hardware: NVIDIA GeForce GTX 1080 Ti with 11G memory is used for most experiments. The linear model is trained locally on a CPU.
Model: mBERT (base, cased) has 12-layer, 768-hidden, 12-heads, 110M parameters. Trained on cased text in the top 104 languages with the largest Wikipedias. XLM-R_B (base) has ~270M parameters with 12-layers, 768-hidden-state, 3072 feed-forward hidden-state, 8-heads, and trained on on 2.5 TB of newly created clean CommonCrawl data in 100 languages. XLM-R_L (large) ~355M parameters with 24-layers, 1027-hidden-state, 4096 feed-forward hidden-state, 16-heads,[9].
Hyperparameters: For the two classification tasks, we use a maximum sequence length of 128. We limit hyperparmeter tuning to a search for the learning rate in $\{7e-6, 1e-5, 3e-5\}$ and use a batch size of 32. For Question Answering, we use a maximum sequence length of 384 with a paragraph slide of 128. We train using a learning rate of $3e-5$ and a batch size of 12 for 2 epochs. The used learning rate for XLM-R_B along with the dev performance for a model with seed=42 is reported in Table 5. We use four other models fine-tuned on the English train split with $seed \in \{2, 4, 8, 16\}$

XNLI	PAWS-X	XQuAD
1e-5	7e-6	3e-5
84.82	92.45	89.10
Accuracy	Accuracy	F1

Table 5: learning rate and English dev performance

Training & Evaluation Runs: Starting from each of the 5 source fine-tuned models, we fine-tune on the target language for 5 different sets of samples. This is repeated for for each size resulting in 25 runs per size. The runtime for the target language fine-tuning varies based on the number of samples used and the number of languages in each dataset. For smaller sample sizes, most runtime is spent for the evaluation on the large test set.

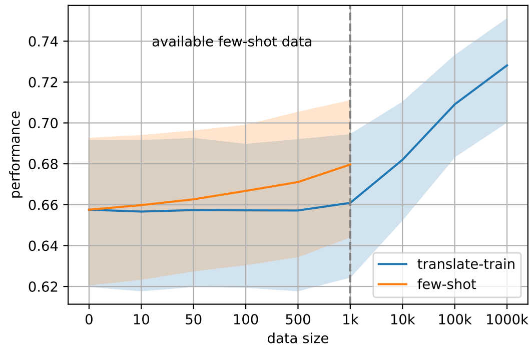
B Languages

code	language properties(1)				cosine similarity to English(2)					XNLI	PAWS-X	XQuAD
	name	size(3)	script	language family	syntax	phonology	inventory	family	geo			
ar	Arabic	1.02	Arabic	Afro-Asiatic	0.65	0.70	0.71	0.00	0.97	x		x
vi	Vietnamese	1.24	Latin	Austro-Asiatic	0.66	0.78	0.75	0.00	0.85	x		x
de	German	2.37	Latin	IE: Germanic	0.90	0.81	0.76	0.54	1.00	x	x	x
en	English	5.98	Latin	IE: Germanic	1.00	1.00	1.00	1.00	1.00	x	x	x
el	Greek	0.17	Greek	IE: Greek	0.78	0.95	0.65	0.15	0.99	x		x
hi	Hindi	0.13	Devanagari	IE: Indo-Aryan	0.62	0.78	0.71	0.13	0.91	x		x
ur	Urdu	0.15	Perso-Arabic	IE: Indo-Aryan	0.62	0.86	0.72	0.13	0.93	x		
es	Spanish	1.56	Latin	IE: Romance	0.82	0.86	0.64	0.10	1.00	x	x	x
ro	Romanian	0.42	0.42	IE: Romance	0.80	0.90	0.73	0.12	0.99			x
fr	French	2.16	Latin	IE: Romance	0.81	0.75	0.74	0.10	1.00	x	x	
ru	Russian	1.58	Cyrillic	IE: Slavic	0.81	0.86	0.65	0.17	0.96	x		x
bg	Bulgarian	0.26	Cyrillic	IE: Slavic	0.86	0.86	0.68	0.14	0.99	x		
ja	Japanese	1.18	Ideograms	Japonic	0.50	0.67	0.65	0.00	0.86		x	
ko	Korean	0.47	Hangul	Koreanic	0.55	0.75	0.71	0.00	0.87		x	
th	Thai	0.13	Brahmic	Kra-Dai	0.64	0.78	0.75	0.00	0.85	x		x
sw	Swahili	0.05	Latin	Niger-Congo	0.46	0.91	0.76	0.00	0.92	x		
zh	Mandarin	1.09	Chinese ideograms	Sino-Tibetan	0.71	0.73	0.70	0.00	0.88	x	x	x
tr	Turkish	0.34	Latin	Turkic	0.51	0.82	0.67	0.00	0.98	x		x

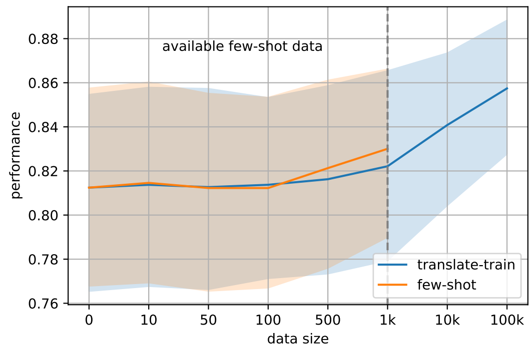
(1) properties taken from XTREME
(2) similarity calculated using lang2vec
(3) size is the #wikipedia articles in millions

Table 6: Languages in the Datasets

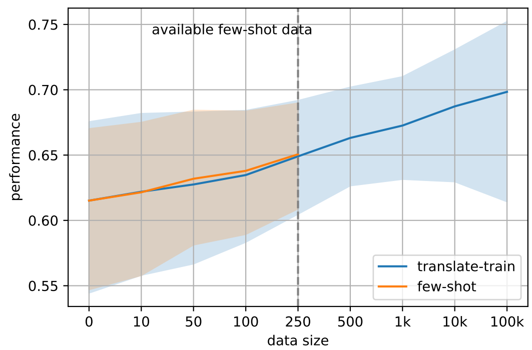
[6] https://github.com/huggingface/transformers
[7] https://github.com/scikit-learn/scikit-learn
[8] https://www.gihtub.com/***
[9] from https://huggingface.co/transformers/pretrained_models.html



(a) Average Accuracy on XNLI across languages

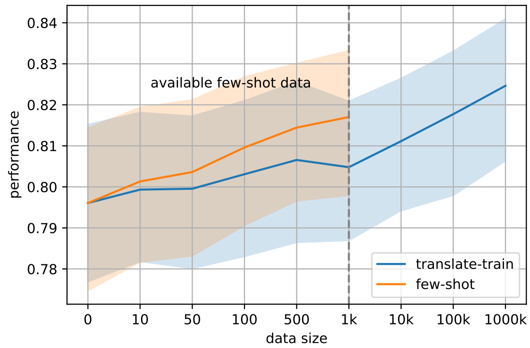


(b) Average Accuracy on PAWS-X across languages

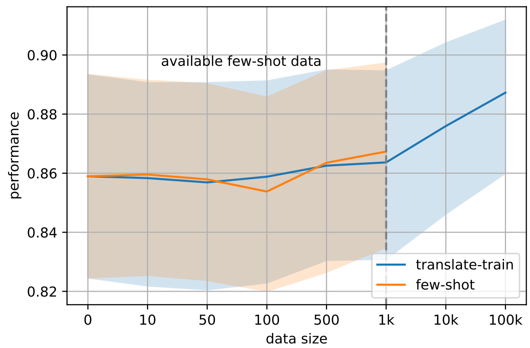


(c) Average F1 on XQuAD across languages

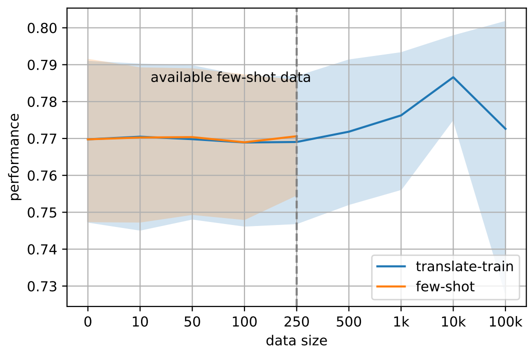
Figure 6: Average performance across languages for **translate-train vs. few-shot** using **mBERT**. The biggest performance boost comes from using translate-train



(a) Average Accuracy on XNLI across languages



(b) Average Accuracy on PAWS-X across languages



(c) Average F1 on XQuAD across languages

Figure 7: Average performance across languages for **translate-train vs. few-shot** using **XLM-R$_L$**. The biggest performance boost comes from using translate-train

C Detailed Results

ar	66.08±0.70	-0.05±0.61	0.18±0.77	1.22±0.36	-0.52±0.67	-0.80±0.91	-0.27±0.52	1.05±0.37	3.61±0.36	5.67±0.38
bg	69.44±0.47	0.15±0.63	0.70±0.73	2.21±0.50	0.03±0.57	0.25±0.88	0.50±0.46	2.23±0.57	4.81±0.64	7.11±0.46
de	71.27±0.61	0.20±0.65	0.84±0.51	1.69±0.63	0.29±0.57	-0.12±1.26	0.25±0.83	1.68±0.46	3.77±0.39	5.50±0.46
el	67.38±0.42	0.21±0.74	0.51±0.96	1.64±0.86	0.08±0.60	-0.31±1.33	-0.51±0.66	1.62±0.70	4.74±0.30	7.38±0.26
es	74.72±0.81	0.04±0.94	0.57±0.92	0.99±0.76	-0.03±0.80	-0.28±1.16	-0.26±0.83	0.88±0.60	2.50±0.66	4.17±0.25
fr	73.70±0.62	0.39±0.58	0.77±0.62	1.18±0.63	-0.01±0.71	-0.19±1.03	-0.48±0.87	0.84±0.50	2.32±0.53	3.97±0.23
hi	60.34±0.41	0.60±0.88	2.04±0.80	3.93±0.41	-0.08±0.74	-0.11±1.45	1.09±0.60	3.41±0.47	6.04±0.44	7.92±0.39
ru	69.28±0.55	0.29±0.69	1.16±0.70	1.66±0.57	-0.05±0.79	0.07±1.09	0.06±0.92	1.33±0.58	3.78±0.41	5.39±0.50
sw	51.31±0.53	-0.05±0.70	0.56±0.37	2.02±0.78	-0.39±0.58	0.21±0.52	0.35±0.62	5.99±0.61	11.14±0.32	13.95±0.41
th	55.05±0.38	0.43±1.01	1.70±0.74	4.37±0.63	-0.63±0.70	-0.86±1.30	0.51±0.99	4.26±0.43	8.77±0.21	10.95±0.17
tr	63.09±0.20	-0.18±0.51	0.17±0.60	1.77±0.44	-0.04±0.68	-0.40±0.75	-0.16±0.62	2.90±0.51	7.10±0.39	9.17±0.41
ur	58.42±0.35	0.26±0.68	1.23±0.97	3.10±0.74	0.48±0.47	1.33±0.75	1.97±0.75	3.39±0.42	4.66±0.42	5.79±0.44
vi	70.62±0.51	0.28±0.78	0.85±1.00	1.82±0.49	-0.13±0.72	0.08±1.20	-0.07±0.72	1.83±0.34	3.94±0.48	5.15±0.44
zh	69.88±0.56	0.55±0.73	1.61±0.68	3.33±0.64	-0.28±0.97	0.66±1.05	1.73±0.70	2.78±0.68	4.96±0.62	6.71±0.35
avg	65.75	0.22	0.92	2.21	-0.09	-0.03	0.33	2.44	5.15	7.06
	0	10	100	1k	10	100	1k	10k	100k	400k
	zer-shot		few-shot				translate-train			

Figure 8: **Detailed Results on XNLI** using **mBERT**. Gains in performance over zero-shot for few-shot and translate-train. Low-resource languages like Swahili have the most gains in both cases

ar	71.98±0.50	0.24±0.79	0.97±0.52	1.33±0.63	-0.22±0.68	0.13±0.99	-0.29±0.83	1.18±0.53	2.70±0.53	3.73±0.45
bg	77.73±0.25	0.36±0.73	1.13±0.43	1.43±0.44	0.42±0.56	0.31±0.80	-0.20±0.73	0.38±0.58	1.30±0.59	2.29±0.39
de	76.59±0.26	0.45±0.64	1.14±0.52	1.84±0.55	0.37±0.63	-0.04±0.74	0.30±0.70	1.19±0.50	2.38±0.26	3.02±0.28
el	76.42±0.42	0.33±0.52	0.56±0.54	1.15±0.42	0.00±0.62	-0.32±1.09	-0.76±0.88	-0.08±0.54	1.65±0.29	2.25±0.34
es	79.02±0.23	0.25±0.52	0.48±0.45	1.18±0.57	0.08±0.68	-0.30±0.99	-0.44±0.75	-0.06±0.42	0.94±0.53	2.20±0.36
fr	78.64±0.57	0.25±0.71	0.65±0.63	0.65±0.65	-0.01±0.73	-0.62±1.26	-0.69±0.87	-0.12±0.60	0.70±0.57	1.85±0.30
hi	70.40±0.96	0.49±1.05	1.35±0.70	2.10±0.66	0.13±1.12	-0.26±1.55	-0.70±0.94	0.82±0.62	2.37±0.54	3.49±0.46
ru	75.99±0.45	0.23±0.54	0.91±0.38	1.38±0.33	-0.06±0.87	0.36±0.68	-0.30±0.88	0.22±0.53	1.68±0.32	2.50±0.36
sw	65.49±0.56	0.04±0.76	0.10±0.80	1.34±0.71	0.17±0.53	-0.70±1.14	-0.83±1.05	1.87±0.55	4.08±0.36	5.42±0.42
th	71.90±0.85	0.31±1.28	1.83±0.49	2.61±0.43	0.55±0.78	0.55±0.97	0.52±0.96	1.77±0.47	3.46±0.60	4.46±0.52
tr	73.17±0.30	0.43±0.61	1.00±0.65	1.47±0.59	0.46±0.84	-0.35±0.97	-0.78±0.80	0.51±0.60	2.07±0.49	3.20±0.42
ur	66.57±0.69	0.76±0.99	2.19±0.56	2.33±0.76	1.26±0.61	1.46±0.98	1.33±0.49	2.13±0.51	1.46±0.61	2.15±0.32
vi	75.39±0.63	0.35±1.02	1.62±0.57	2.18±0.54	0.40±0.85	0.28±1.10	0.13±0.67	1.08±0.51	2.34±0.48	3.57±0.31
zh	73.75±0.48	0.52±0.75	1.91±0.63	2.97±0.51	0.56±0.94	0.29±1.13	0.69±0.85	1.67±0.71	2.88±0.42	4.15±0.48
avg	73.79	0.36	1.13	1.71	0.29	0.06	-0.14	0.90	2.14	3.16
	0	10	100	1k	10	100	1k	10k	100k	400k
	zer-shot		few-shot				translate-train			

Figure 9: **Detailed Results on XNLI** using **XLM-R**$_B$. Gains in performance over zero-shot for few-shot and translate-train. Low-resource languages like Swahili have the most gains in both cases

ar	78.77±0.73	0.48±0.70	1.65±0.52	2.30±0.57	0.03±0.93	0.94±0.78	1.29±0.42	1.57±0.29	2.47±0.28	3.76±0.39
bg	83.75±0.46	0.67±0.45	1.23±0.31	1.69±0.25	0.25±0.52	0.89±0.32	0.70±0.45	0.99±0.32	1.70±0.48	2.17±0.29
de	82.87±0.47	0.36±0.58	0.71±0.49	1.41±0.28	0.08±0.55	0.06±0.71	0.75±0.34	0.99±0.30	1.54±0.38	2.31±0.33
el	82.37±0.71	0.51±0.60	1.22±0.30	1.82±0.43	0.18±0.62	0.73±0.56	0.64±0.68	1.16±0.45	2.19±0.38	2.69±0.25
es	84.59±0.46	0.33±0.45	0.70±0.50	1.29±0.40	0.09±0.57	0.30±0.67	0.20±0.63	0.59±0.45	1.26±0.41	1.61±0.39
fr	83.26±0.24	0.42±0.32	0.95±0.29	1.42±0.26	0.11±0.55	0.33±0.66	0.38±0.56	0.80±0.31	1.52±0.19	2.13±0.22
hi	76.83±0.88	0.54±0.93	1.48±0.56	2.67±0.26	0.53±0.72	0.90±0.84	1.03±0.47	2.20±0.37	2.52±0.33	3.29±0.37
ru	80.88±0.45	0.55±0.48	1.31±0.34	2.00±0.27	0.26±0.62	1.01±0.41	0.69±0.77	0.99±0.50	2.17±0.44	2.37±0.18
sw	72.33±0.85	0.28±0.95	1.14±0.64	2.10±0.75	0.37±0.82	0.11±0.81	0.21±1.14	2.54±0.46	4.05±0.15	5.01±0.22
th	77.18±1.23	0.85±1.04	2.39±0.58	2.96±0.45	0.61±1.07	1.26±1.20	1.71±0.76	2.40±0.48	3.22±0.48	4.16±0.28
tr	79.35±0.52	0.34±0.60	1.20±0.38	1.79±0.38	0.31±0.57	0.20±0.84	0.55±0.46	1.36±0.46	1.84±0.29	2.74±0.43
ur	72.56±1.16	1.09±1.07	2.15±0.56	3.12±0.46	1.02±1.03	1.98±0.78	1.77±0.63	2.20±0.47	0.99±0.44	1.10±0.34
vi	80.12±0.65	0.40±0.81	1.36±0.37	2.22±0.32	0.28±0.58	0.42±0.68	0.87±0.34	1.47±0.28	2.11±0.32	2.99±0.32
zh	79.64±0.89	0.53±0.76	1.47±0.44	2.55±0.26	0.47±0.71	0.70±0.87	1.43±0.51	1.84±0.39	2.73±0.35	3.70±0.22
avg	79.61	0.53	1.35	2.10	0.33	0.70	0.87	1.51	2.17	2.86
	0	10	100	1k	10	100	1k	10k	100k	400k
	zer-shot		few-shot				translate-train			

Figure 10: **Detailed Results on XNLI** using **XLM-R**$_L$. Gains in performance over zero-shot for few-shot and translate-train. Low-resource languages like Swahili have the most gains in both cases

	0	10	100	1k	10	100	1k	10k	50k
de	84.86±1.27	0.23±1.17	0.12±1.54	1.97±0.98	0.20±1.07	0.36±0.85	0.47±0.88	1.82±0.98	2.70±0.66
es	87.87±1.31	0.04±1.06	-0.04±1.18	0.71±0.96	-0.06±1.27	-0.28±1.22	0.26±0.86	1.53±0.64	2.62±0.52
fr	87.03±0.64	0.07±0.63	-0.12±0.90	0.67±0.54	-0.03±0.57	-0.04±0.60	0.39±0.35	1.40±0.38	2.78±0.56
ja	74.23±2.36	0.14±1.54	-0.28±2.15	2.76±1.28	-0.14±1.58	0.20±1.62	1.91±0.90	5.02±0.73	7.42±1.07
ko	74.51±2.31	0.75±1.43	0.15±2.28	2.28±1.43	0.55±1.80	0.20±1.56	1.04±1.40	3.73±1.29	6.30±1.26
zh	78.96±2.22	0.08±1.37	0.05±1.71	2.18±1.25	0.25±1.25	0.34±1.11	1.75±0.94	3.53±0.59	5.18±0.48
avg	81.24	0.22	-0.02	1.76	0.13	0.13	0.97	2.84	4.50

Figure 11: **Detailed Results on PAWS-X** using **mBERT**. Gains in performance over zero-shot for few-shot and translate-train. Non-European language show the most gain especially Japanese and Korean.

	0	10	100	1k	10	100	1k	10k	50k
de	86.75±0.95	-0.34±0.97	-0.42±1.18	0.17±0.74	-0.29±1.00	-0.33±1.02	-0.52±1.12	0.46±1.00	1.29±0.82
es	87.94±0.65	0.02±0.53	-0.52±1.18	0.24±0.55	-0.82±0.97	-0.91±1.17	-0.01±0.64	0.77±0.70	1.77±0.38
fr	88.74±0.85	-0.16±0.73	-0.18±0.86	0.10±0.65	-0.07±0.58	-0.59±0.93	0.11±0.57	0.68±0.56	1.58±0.57
ja	75.91±0.59	0.07±0.56	-1.05±1.60	0.63±0.87	0.10±0.51	0.10±0.85	0.34±0.85	1.96±0.68	3.31±0.80
ko	73.95±1.32	1.02±0.93	0.85±0.77	1.92±0.88	-0.18±1.75	0.81±0.94	0.96±1.03	4.05±0.78	6.43±1.07
zh	79.16±1.43	0.52±0.66	0.11±1.13	1.49±0.63	0.20±1.43	0.07±1.08	1.14±1.00	2.65±0.57	3.54±0.59
avg	82.07	0.19	-0.20	0.76	-0.18	-0.14	0.34	1.76	2.99

Figure 12: **Detailed Results on PAWS-X** using **XLM-R**$_B$. Gains in performance over zero-shot for few-shot and translate-train. Non-European language show the most gain especially Korean.

	0	10	100	1k	10	100	1k	10k	50k
de	90.26±0.25	-0.56±0.86	-0.78±0.87	0.37±0.54	-0.21±0.59	-0.48±0.60	-0.27±0.45	0.46±0.35	1.25±0.17
es	90.47±0.47	-0.05±0.88	-0.75±1.32	0.46±0.48	-0.56±1.15	-0.33±1.30	0.17±0.57	1.06±0.34	1.69±0.47
fr	90.76±0.53	-0.31±0.99	-0.73±0.91	0.04±0.51	-0.23±0.76	-0.06±0.62	0.07±0.45	0.93±0.42	1.65±0.39
ja	80.36±0.76	0.41±0.42	-0.72±1.56	1.02±0.58	0.23±1.00	-0.03±1.29	0.55±0.53	2.33±0.57	4.00±0.29
ko	80.31±1.55	0.70±1.57	0.34±1.77	2.11±0.64	0.34±1.65	0.79±1.11	1.50±0.77	3.36±0.36	4.94±0.24
zh	83.21±0.82	0.16±1.01	-0.46±1.67	1.03±0.88	0.06±0.98	0.03±0.85	0.79±0.64	2.06±0.38	3.46±0.35
avg	85.89	0.06	-0.52	0.84	-0.06	-0.01	0.47	1.70	2.83

Figure 13: **Detailed Results on PAWS-X** using **XLM-R**$_L$. Gains in performance over zero-shot for few-shot and translate-train. Non-European language show the most gain especially Japanese and Korean.

	zer-shot 0	few-shot 10	few-shot 100	few-shot 250	translate-train 10	translate-train 100	translate-train 250	translate-train 1k	translate-train 10k	translate-train 88k
ar	61.36±0.80	0.54±1.17	1.47±1.65	2.20±0.81	0.43±0.89	1.46±1.27	2.39±0.90	4.83±0.87	7.73±0.58	9.17±1.03
de	72.06±1.02	0.18±0.69	0.23±0.95	0.10±0.83	0.35±0.82	0.00±0.92	0.03±0.87	0.81±0.88	2.59±0.52	4.20±0.64
el	61.92±1.36	0.60±1.12	2.26±1.23	2.61±1.14	0.14±1.38	1.64±1.40	2.07±1.10	4.65±0.86	7.42±0.59	10.60±0.95
es	74.38±0.74	0.37±1.05	0.63±0.92	0.65±0.63	0.59±0.96	0.86±1.03	1.37±0.90	2.76±0.58	4.04±0.50	5.80±0.42
hi	56.41±1.11	0.42±1.10	2.43±1.27	3.82±1.06	0.76±1.24	2.02±1.17	3.90±1.30	7.21±0.67	11.70±0.73	15.11±0.63
ru	70.76±0.50	-0.13±0.77	-0.59±1.03	-0.59±0.93	0.38±0.39	0.15±0.75	0.26±0.73	1.46±0.61	2.84±0.56	5.05±0.60
th	38.16±0.52	2.98±1.66	9.69±1.92	15.40±0.85	2.13±1.65	7.39±2.13	12.77±1.24	17.18±1.68	8.89±2.12	-0.18±1.50
tr	52.95±1.03	1.25±1.24	4.45±1.46	6.68±1.15	1.73±1.04	4.70±1.63	7.29±1.46	10.98±1.10	13.90±0.66	16.82±0.73
vi	69.16±0.43	0.31±0.94	1.44±0.97	2.63±0.86	0.27±1.05	1.27±1.01	2.22±1.02	3.94±0.71	6.28±0.61	8.31±0.27
zh	57.98±0.85	-0.15±1.01	0.81±1.36	1.99±1.18	0.03±0.87	0.09±1.36	1.59±1.01	3.64±0.95	6.76±0.64	8.39±0.39
avg	61.51	0.64	2.28	3.55	0.68	1.96	3.39	5.75	7.22	8.33

Figure 14: **Detailed Results on XQuAD** using **mBERT**. Gains in performance over zero-shot for few-shot and translate-train. Non-European languages show the most gain especially Chinese. Thai shows a significant degrade when using the full machine-translated dataset. This might be due to lower-quality machine translation for Thai.

	zer-shot 0	few-shot 10	few-shot 100	few-shot 250	translate-train 10	translate-train 100	translate-train 250	translate-train 1k	translate-train 10k	translate-train 88k
ar	67.76±0.61	0.08±0.57	-0.32±0.86	-0.29±0.81	0.29±0.52	-0.03±0.66	0.17±0.75	1.49±0.74	3.68±0.81	3.31±0.37
de	74.75±1.02	-0.26±0.90	-0.70±1.00	-1.31±0.85	0.03±0.87	-0.67±0.97	-1.05±1.00	-1.04±0.90	0.22±0.55	1.00±0.29
el	73.01±0.32	-0.36±0.37	-0.75±0.75	-0.99±0.54	-0.35±0.45	-1.06±0.88	-1.29±0.56	-0.89±0.60	0.27±0.63	2.36±0.41
es	76.16±0.70	-0.13±0.52	-0.45±0.54	-0.53±0.64	-0.13±0.57	-0.53±0.64	-0.16±0.72	0.35±0.75	1.07±0.58	2.88±0.16
hi	68.36±1.17	-0.00±0.84	-0.22±0.83	-0.40±0.68	0.23±0.95	-0.13±0.80	0.20±0.91	0.98±0.89	3.25±0.64	4.73±0.44
ru	73.53±0.96	-0.10±0.86	-0.76±0.76	-1.08±0.64	-0.04±0.84	-0.26±0.77	-0.58±0.76	-0.38±0.77	0.72±0.52	2.74±0.86
th	66.40±1.08	0.79±0.92	1.58±0.88	2.53±1.12	0.67±1.09	1.72±1.09	3.36±1.35	7.24±0.81	1.73±1.63	-15.94±1.56
tr	67.11±1.19	-0.05±1.14	-0.13±1.33	-0.34±0.66	-0.06±0.95	-0.17±0.69	0.22±1.05	1.44±0.89	3.61±0.65	4.22±0.65
vi	73.84±0.33	-0.04±0.43	0.04±0.77	0.39±0.94	0.05±0.41	-0.02±0.71	-0.20±0.63	0.73±0.71	2.27±0.43	3.39±0.35
zh	64.19±0.94	0.46±0.76	1.51±0.94	3.75±1.08	-0.39±0.85	-0.54±1.20	0.09±1.08	1.95±0.89	9.88±0.79	7.78±1.84
avg	70.51	0.04	-0.02	0.17	0.03	-0.17	0.08	1.19	2.67	1.65

Figure 15: **Detailed Results on XQuAD** using **XLM-R**$_B$. Gains in performance over zero-shot for few-shot and translate-train. Non-European languages show the most gain especially Chinese. Thai shows a significant degrade when using the full machine-translated dataset. This might be due to lower-quality machine translation for Thai.

	zer-shot 0	few-shot 10	few-shot 100	few-shot 250	translate-train 10	translate-train 100	translate-train 250	translate-train 1k	translate-train 10k	translate-train 88k
ar	76.67±0.26	0.38±0.37	-0.12±0.58	-0.50±0.82	0.50±0.46	0.42±0.74	0.24±0.75	0.98±0.55	2.16±0.58	1.38±0.45
de	80.00±0.78	-0.03±0.56	-0.62±0.68	-0.89±0.77	-0.17±0.63	-0.74±0.73	-0.89±0.77	-1.23±0.83	-0.71±0.56	-0.01±0.57
el	79.51±0.67	-0.20±0.49	-0.59±0.68	-0.89±0.73	-0.12±0.49	-0.94±0.67	-1.07±0.66	-0.91±0.62	-0.25±0.65	0.35±0.88
es	80.91±0.63	0.09±0.66	-0.08±0.60	-0.31±0.59	0.06±0.71	-0.25±0.74	-0.28±0.52	0.30±0.65	1.39±0.68	2.11±0.20
hi	75.85±0.32	0.10±0.39	-0.33±0.81	-0.66±0.85	0.16±0.47	-0.16±0.55	-0.23±0.74	0.05±0.72	0.89±0.54	1.98±0.55
ru	79.55±0.65	-0.13±0.45	-0.97±0.61	-1.17±0.61	-0.07±0.51	-0.48±0.59	-0.66±0.54	-0.83±0.54	-0.17±0.55	0.69±0.45
th	73.86±1.20	0.38±0.72	0.87±1.16	2.00±1.00	0.59±1.05	1.73±0.82	2.77±0.60	6.38±0.59	2.00±1.42	-14.07±1.49
tr	74.08±0.53	-0.02±0.41	0.03±0.47	-0.04±0.72	0.10±0.40	0.08±0.65	0.13±0.59	0.93±0.72	2.39±0.71	2.94±0.98
vi	79.93±0.49	-0.11±0.42	-0.06±0.55	0.04±0.51	-0.15±0.44	-0.17±0.60	-0.53±0.55	-0.31±0.45	0.46±0.54	1.20±0.41
zh	69.41±0.44	0.03±0.66	1.06±1.21	3.27±1.69	-0.17±0.54	-0.34±0.92	-0.19±0.93	1.13±0.82	8.67±1.49	6.29±0.95
avg	76.98	0.05	-0.08	0.08	0.07	-0.09	-0.07	0.65	1.68	0.29

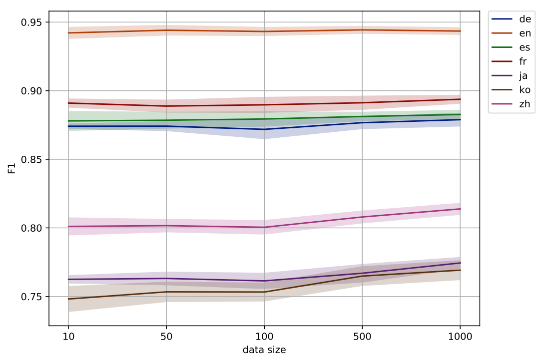
Figure 16: **Detailed Results on XQuAD** using **XLM-R**$_L$. Gains in performance over zero-shot for few-shot and translate-train. Non-European languages show the most gain especially Chinese. Thai shows a significant degrade when using the full machine-translated dataset. This might be due to lower-quality machine translation for Thai.

	0	10	100	1k	10	100	1k	10k	100k	400k
	zer-shot		few-shot				translate-train			
ar	71.98±0.50	0.44±1.06	1.15±0.70	1.44±0.80	-2.15±2.34	-0.11±1.12	0.40±0.86	0.82±1.14	1.77±0.21	4.29±0.47
bg	77.73±0.25	0.46±0.99	1.07±0.36	1.39±0.48	0.09±1.36	0.18±0.83	-0.69±1.68	0.55±0.45	1.87±0.32	1.82±0.48
de	76.59±0.26	0.58±0.99	1.05±0.52	1.91±0.56	0.40±1.02	-0.15±0.96	0.06±1.41	1.17±0.51	1.69±0.68	2.93±0.48
el	76.42±0.42	0.06±0.99	0.53±0.75	1.28±0.47	-0.31±1.20	0.02±0.80	-1.10±1.55	0.23±0.61	0.88±0.27	0.79±0.23
es	79.02±0.23	0.24±0.92	0.30±0.64	1.06±0.57	-0.04±0.72	-0.41±0.86	-1.15±1.07	0.26±0.65	0.65±0.49	1.81±0.26
fr	78.64±0.57	0.24±0.85	0.32±0.73	0.77±0.64	-0.31±1.09	-0.65±1.20	-0.35±1.07	-0.08±0.91	1.03±0.41	1.42±0.27
hi	70.40±0.96	0.49±1.40	1.31±0.94	1.98±0.73	-0.38±1.37	-0.05±1.38	-1.07±1.59	0.97±1.23	2.68±0.26	2.76±0.56
ru	75.99±0.45	0.17±0.81	0.84±0.46	1.21±0.28	-0.43±1.10	-0.09±0.75	-0.15±0.61	0.25±0.87	1.39±0.29	1.45±1.07
sw	65.49±0.56	-0.10±0.83	0.53±0.99	1.32±0.89	-0.03±0.81	-0.73±1.21	-0.37±1.56	2.38±0.63	3.50±0.33	4.87±0.84
th	71.90±0.85	0.79±1.68	2.17±0.36	2.72±0.63	0.07±1.64	0.62±1.24	1.04±0.70	2.14±0.73	3.66±0.08	4.22±0.37
tr	73.17±0.30	-0.02±1.20	1.07±0.68	1.44±0.62	0.43±1.03	-0.08±0.95	-0.50±1.06	0.89±0.88	1.52±0.45	1.97±0.44
ur	66.57±0.69	0.85±1.56	1.91±0.68	2.51±0.50	0.07±1.09	0.72±0.67	0.80±0.66	0.21±0.81	-0.43±0.29	0.49±0.46
vi	75.39±0.63	0.92±1.51	1.71±0.62	2.03±0.67	0.40±1.11	0.53±0.98	-0.11±1.19	1.31±0.74	2.22±0.31	3.24±0.27
zh	73.75±0.48	0.70±1.45	2.13±0.48	3.00±0.48	-0.44±1.52	-0.13±1.23	0.56±1.51	2.06±0.92	2.73±0.48	3.61±0.31
avg	73.79	0.41	1.15	1.72	-0.19	-0.02	-0.19	0.94	1.80	2.55

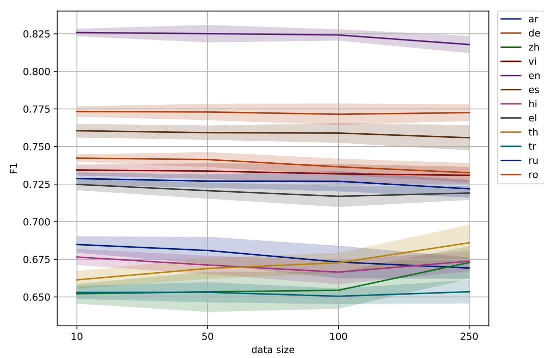
Figure 17: Detailed Results on **XNLI using a part of the available data as dev**. The few-shot performance only changes slightly with minor increases and decreases for across the languages. The highest increase on average is at 10 samples with an increase of 0.05%. Translate-train performance decreases for almost all languages and on average.

	0	10	100	1k	10	100	1k	10k	50k
de	86.75±0.95	-1.20±1.58	-0.25±0.71	0.22±0.75	-0.64±1.42	-1.12±0.97	-0.85±1.22	0.17±1.04	1.69±0.42
es	87.94±0.65	-0.57±0.97	0.08±0.51	0.26±0.46	-1.40±1.44	-0.31±1.06	-0.56±0.85	0.64±0.54	2.53±0.60
fr	88.74±0.85	-0.77±1.33	-0.27±0.74	-0.08±0.78	-0.82±1.39	-0.61±1.17	-0.30±0.85	0.29±1.07	2.49±0.38
ja	75.91±0.59	-0.56±1.29	0.05±0.55	0.26±1.07	-0.77±1.38	-0.44±0.94	0.03±1.13	2.13±0.82	5.42±0.40
ko	73.95±1.32	-0.33±1.65	1.09±1.04	2.19±0.76	-0.26±2.11	0.49±1.39	1.23±1.57	4.27±1.05	7.71±0.64
zh	79.16±1.43	0.53±0.79	0.55±0.95	1.20±0.85	-0.10±1.47	-0.12±1.32	0.61±1.02	2.42±0.53	4.71±0.18
avg	82.07	-0.48	0.21	0.67	-0.67	-0.35	0.03	1.65	4.09

Figure 18: Detailed Results on **PAWS-X using a part of the available data as dev**. The few-shot performance shows mixed gains decreasing by ∼0.60% for 10 samples, increasing by ∼0.40% at 100 then decreasing againg by ∼0.10%. Translate-train performance decreases util the full dataset is used where it increases by ∼1%.



(a) **PAWS-X Performance variance on different shots.** Variance decreases with an increased data size



(b) **XQuAD Performance variance on different shots.** Variance increases with an increased data size

Figure 19: Performance variance on different shots

lang	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	avg
10	0.64	0.42	0.42	0.51	0.63	0.75	0.49	-0.25	0.63	0.48	0.70	0.19	0.48	0.42	0.50
50	0.82	0.59	0.59	0.76	0.74	0.87	0.66	0.11	0.70	0.81	0.84	0.54	0.73	0.56	0.69
100	0.76	0.53	0.47	0.46	0.64	0.77	0.69	-0.50	0.58	0.58	0.64	0.06	0.73	0.33	0.52
500	0.84	0.64	0.77	0.79	0.73	0.81	0.77	0.18	0.67	0.71	0.88	0.54	0.77	0.73	0.72
1000	0.72	0.63	0.74	0.69	0.72	0.84	0.60	0.10	0.06	0.51	0.80	0.03	0.51	0.75	0.58
all	0.77	0.59	0.62	0.69	0.73	0.79	0.66	0.15	0.62	0.57	0.79	0.38	0.65	0.55	0.64

Table 7: **XNLI Pearson correlation** between the performance on **English** and the performance on other languages using the same set of samples.

	de	en	es	fr	ja	ko	zh
de	1.00	0.66	0.52	0.56	0.21	0.54	0.64
en	0.66	1.00	0.56	0.41	0.11	0.37	0.36
es	0.52	0.56	1.00	0.57	0.22	0.54	0.57
fr	0.56	0.41	0.57	1.00	0.03	0.59	0.55
ja	0.21	0.11	0.22	0.03	1.00	0.16	0.32
ko	0.54	0.37	0.54	0.59	0.16	1.00	0.54
zh	0.64	0.36	0.57	0.55	0.32	0.54	1.00
avg	0.59	0.50	0.57	0.53	0.29	0.54	0.57

Table 8: **PAWS-X Pearson correlation** of the performance between languages.

lang	de	es	fr	ja	ko	zh	avg
10	0.47	0.65	0.34	-0.22	0.53	0.56	0.48
50	0.81	0.56	0.57	-0.35	0.53	0.48	0.51
100	0.78	0.53	0.42	0.40	0.47	0.44	0.57
500	0.52	0.55	0.53	0.16	0.41	0.11	0.47
1000	0.75	0.77	0.30	-0.01	-0.02	0.35	0.45
all	0.66	0.56	0.41	0.11	0.37	0.36	0.50

Table 9: **PAWS-X Pearson correlation** between the performance on **English** and the performance on other languages using the same set of samples.

de	es	fr	ja	ko	zh
0.66	0.62	0.68	0.45	0.38	0.52

Table 10: **PAWS-X Pearson correlation** between the performance of **machine translation** and manual translation.

	ar	de	zh	vi	en	es	hi	el	th	tr	ru	ro
ar	1.00	-0.14	0.03	0.07	0.12	-0.02	0.01	-0.03	0.07	0.25	0.12	-0.06
de	-0.14	1.00	-0.54	-0.18	0.35	0.57	0.42	0.22	-0.26	0.40	-0.09	-0.00
zh	0.03	-0.54	1.00	0.16	-0.37	-0.38	-0.21	-0.41	0.55	-0.17	-0.24	-0.22
vi	0.07	-0.18	0.16	1.00	-0.08	-0.02	-0.08	-0.01	0.02	-0.18	-0.12	-0.26
en	0.12	0.35	-0.37	-0.08	1.00	0.46	0.08	0.07	-0.17	0.06	-0.04	-0.06
es	-0.02	0.57	-0.38	-0.02	0.46	1.00	0.10	0.02	-0.31	0.09	-0.29	-0.24
hi	0.01	0.42	-0.21	-0.08	0.08	0.10	1.00	0.18	0.06	0.37	0.27	0.18
el	-0.03	0.22	-0.41	-0.01	0.07	0.02	0.18	1.00	-0.15	0.01	0.34	0.13
th	0.07	-0.26	0.55	0.02	-0.17	-0.31	0.06	-0.15	1.00	0.17	0.07	0.10
tr	0.25	0.40	-0.17	-0.18	0.06	0.09	0.37	0.01	0.17	1.00	0.33	0.27
ru	0.12	-0.09	-0.24	-0.12	-0.04	-0.29	0.27	0.34	0.07	0.33	1.00	0.56
ro	-0.06	-0.00	-0.22	-0.26	-0.06	-0.24	0.18	0.13	0.10	0.27	0.56	1.00
avg	0.12	0.15	-0.07	0.03	0.12	0.08	0.20	0.11	0.10	0.22	0.16	0.12

Table 11: **XQuAD Pearson correlation** of the performance between languages.

lang	ar	de	zh	vi	es	hi	el	th	tr	ru	ro	avg
10	0.54	0.41	0.08	-0.40	0.30	0.05	-0.10	0.43	0.44	-0.25	-0.23	0.19
50	0.37	0.24	-0.28	0.11	-0.01	0.19	0.27	0.12	0.21	-0.04	-0.08	0.18
100	-0.37	0.35	-0.54	-0.03	0.71	0.02	0.08	-0.09	-0.08	-0.40	-0.12	0.05
250	0.08	0.20	-0.25	0.03	0.65	-0.16	-0.38	-0.31	-0.45	-0.33	-0.34	-0.02
all	0.12	0.35	-0.37	-0.08	0.46	0.08	0.07	-0.17	0.06	-0.04	-0.06	0.12

Table 12: **XQuAD Pearson correlation** between the performance on **English** and the performance on other languages using the same set of samples.

	XNLI		PAWS-X	
	MSE	RMSE	MSE	RMSE
avg (baseline)	1.05±0.56	0.99±0.26	1.26±0.76	1.08±0.34
model using features:				
en performance	0.68±0.41	0.80±0.23	1.08±0.92	0.97±0.42
mt performance	0.34±0.28	0.56±0.20	0.92±0.56	0.93±0.28
en + mt performance	0.33±0.26	0.55±0.18	0.91±0.56	0.92±0.28
+ lang features	**0.32±0.25**	**0.54±0.18**	**0.58±0.27**	**0.75±0.17**
only lang features	0.93±0.47	0.93±0.24	1.01±0.45	0.98±0.25

Table 13: **Performance prediction error**. Predicting the few-shot performance gain using models with the English and MT performance as features. For each language, the average performance gain for all other languages is used as a baseline. *+lang features* further adds features from lang2vec. Having a combination of English and MT performance with language features achieves the best results.

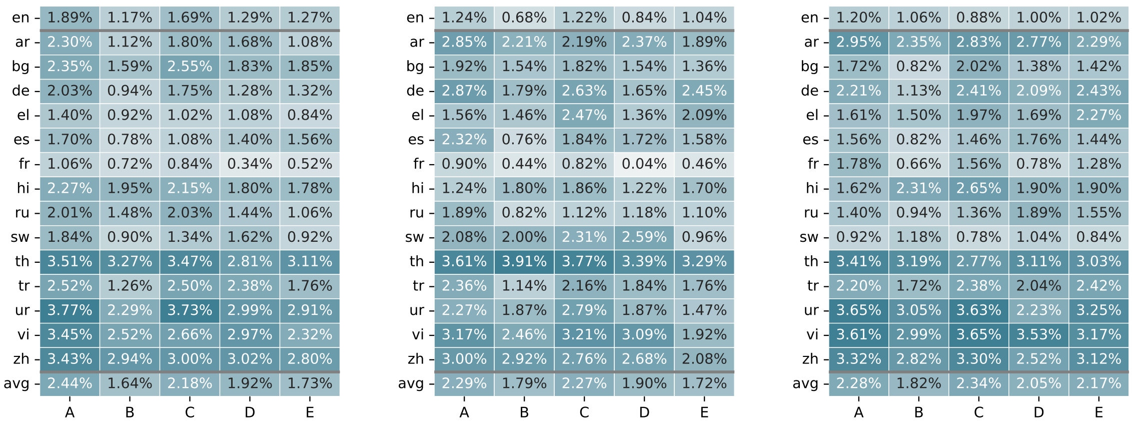
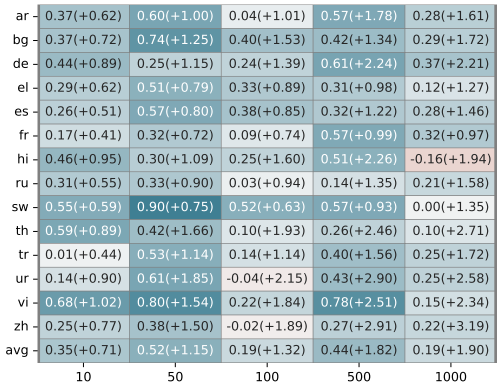
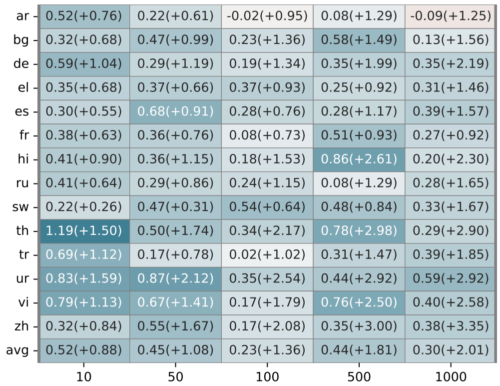


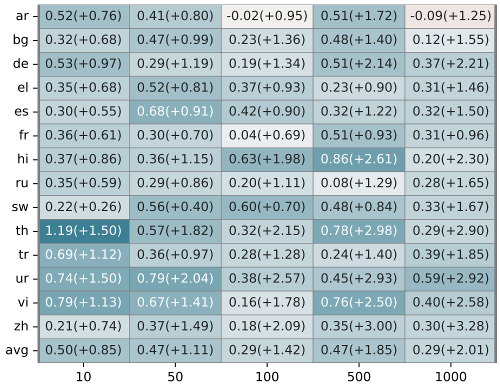
Figure 20: **XNLI few-shot gain** over zero-shot across 5 sets of samples (**size=1000**) for 3 different model initalizations. Sets A and C yield better performance for the 3 different initalizations. The English performance can be used as an indicator.



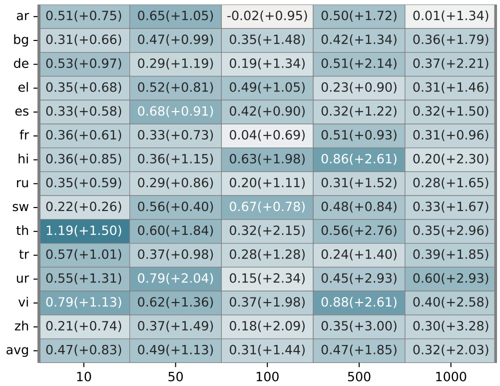
(a) XNLI chosen-shots gain
using English performance



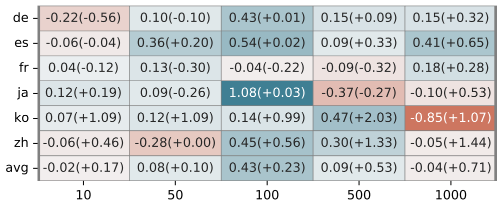
(b) XNLI chosen-shots gain
using machine translation performance



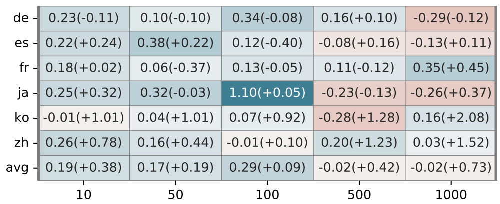
(c) XNLI chosen-shots gain
using (en + mt) model



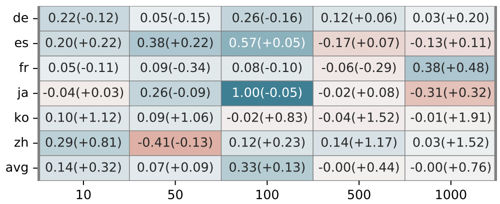
(d) XNLI chosen-shots gain
using (en + mt + lang features) model



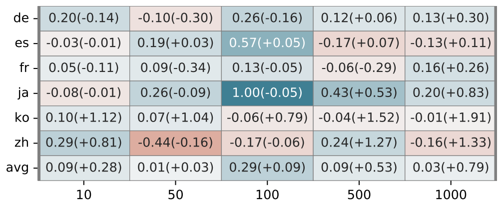
(e) PAWSX chosen-shots gain
using English performance



(f) PAWS-X chosen-shots gain
using machine translation performance

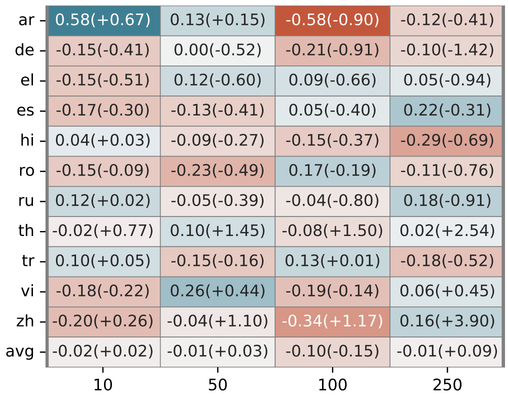


(g) PAWS-X chosen-shots gain
using (en + mt) model

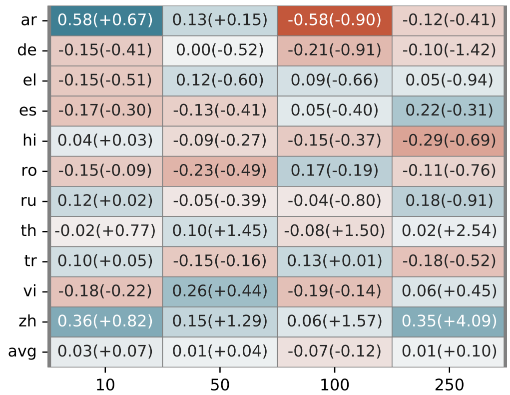


(h) PAWS-X chosen-shots gain
using (en + mt + lang features) model

Figure 21: **Chosen-shots gain in performance**. The gain of choosing shots over the average of no-choosing (average over 5 random sets). The actual few-shot gain (compared to zero-shot) is shown in parenthesis as follows: *chosen-shots-gain (few-shot-gain)*. When *chosen-shots-gain* is positive (green), choosing the shots results in more gain. When negative (red), it hurts and results in less gain.



(a) XQuAD chosen-shots gain using English performance



(b) XQuAD chosen-shots gain using en performance model

Figure 22: **XQuAD chosen-shots gain in performance** (no gain!). The gain of choosing shots over the average of no-choosing (average over 5 random sets). The actual few-shot gain (compared to zero-shot) is shown in parenthesis as follows chosen-shots-gain (few-shot-gain). We can see that there is no gain in choosing the shots. Experiments with adding language features to the model further decrease the performance.