Training Dynamics for Curriculum Learning: A Study on Monolingual and Cross-lingual NLU

Anonymous ACL submission

Abstract

Curriculum Learning (CL) is a technique of training models via ranking examples in a typically increasing difficulty trend with the aim of accelerating convergence and improving generalisability. Current approaches for Natural Language Understanding (NLU) tasks use CL to improve in-distribution data performance often via heuristic-oriented difficulties or task-agnostic ones. In this work, instead, we employ CL for NLU by taking advantage of training dynamics as difficulty metrics, i.e. statistics that measure the behavior of the model at hand on specific task-data instances during training and propose modifications of existing CL schedulers based on these statistics. Differently from existing works, we focus on evaluating models on in-distribution, out-of-distribution as well as zero-shot cross-lingual transfer datasets. We show across several NLU tasks that CL with training dynamics can result in better performance mostly on zero-shot cross-lingual transfer and OOD settings with improvements up by 8.5%. Overall, experiments indicate that training dynamics can lead to better performing models with smoother training compared to other difficulty metrics while at the same time being up to 51% faster. In addition, through analysis we shed light on the correlations of task-specific versus task-agnostic metrics[1].

1 Introduction

Transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019, LMs) have recently achieved great success in a variety of NLP tasks (Wang et al., 2018, 2019a). However, generalisation to out-of-distribution (OOD) data and zero-shot cross-lingual transfer still remain a challenge (Linzen, 2020; Hu et al., 2020). Among existing techniques, improving OOD performance has been addressed by training with adversarial data (Yi et al., 2021), while better transfer across languages has mostly focused on selecting appropriate languages to transfer from (Lin et al., 2019; Turc et al., 2021), has employed meta-learning (Nooralahzadeh et al., 2020) or data alignment (Fang et al., 2020).

Contrastive to such approaches that take advantage of additional training data is Curriculum Learning (Bengio et al., 2009, CL), a technique that aims to train models using a specific ordering of the original training examples. This ordering typically follows an increasing difficulty trend where easy examples are fed to the model first, moving towards harder instances. The intuition behind CL stems from human learning, as humans focus on simpler concepts before learning more complex ones, a procedure that is called shaping (Krueger and Dayan, 2009). Although curricula have been primarily used for Computer Vision (Hacohen and Weinshall, 2019; Wu et al., 2021) and Machine Translation (Zhang et al., 2019a; Platanios et al., 2019), there are only a handful of approaches that incorporate CL into Natural Language Understanding tasks (Sachan and Xing, 2016; Tay et al., 2019; Lalor and Yu, 2020; Xu et al., 2020a).

Typically, CL requires a measure of difficulty for each example in the training set. Existing methods using CL in NLU tasks rely on heuristics such as sentence length, word rarity, depth of the dependency tree (Platanios et al., 2019; Tay et al., 2019), metrics based on item-response theory (Lalor and Yu, 2020) or task-agnostic model metrics such as perplexity (Zhou et al., 2020). Such metrics have been employed to either improve in-distribution performance on NLU or Machine Translation. However, their effect is still under-explored on other settings.

In this study instead, we propose to adopt Training dynamics (Swayamdipta et al., 2020, TD) as difficulty measures for CL and fine-tune models with curricula on downstream tasks. TD were recently proposed as a set of statistics collected dur-

[1] Code will be made available upon acceptance.

ing the course of a model's training to automatically evaluate dataset quality, by identifying annotation artifacts. These statistics, offer a 3-dimensional view of a model's uncertainty towards each training example classifying them into distinct areas–*easy*, *ambiguous* and *hard* examples for a model to learn.

We test a series of easy-to-hard curricula using TD with existing schedulers as well as novel modifications of those and experiment with other task-specific and task-agnostic metrics. We show performances and training times on three settings: in-distribution (ID), out-of-distribution (OOD) and zero-shot (ZS) transfer to languages different than English. To the best of our knowledge, no prior work on NLU considers the impact of CL on all these settings. To consolidate our findings, we evaluate models on different classification tasks, including Natural Language Inference, Paraphrase Identification, Commonsense Causal Reasoning and Document Classification.

Our findings suggest that TD-CL provides better zero-shot cross-lingual transfer up to 1.2% over prior work and can gain speedups up to 51%. In ID settings CL has minimal impact, while in OOD settings models trained with TD-CL can boost performance up to 8.5%. over prior work. Finally, TD provide more stable training compared to another task-specific metric. On the other hand, heuristics can also offer improvements particularly when testing on a completely different domain.

2 Related Work

Curriculum Learning was initially mentioned in the work of Elman (1993) who demonstrated the importance of feeding neural networks with small/easy inputs at the early stages of training. The concept was later formalised by Bengio et al. (2009) where training in an easy-to-hard ordering was shown to result in faster convergence and improved performance. In general, Curriculum Learning requires a *difficulty metric* (also known as the scoring function) used to rank training instances, and a *scheduler* (known as the pacing function) that decides when and how new examples–of different difficulty–should be introduced to the model.

Example Difficulty was initially expressed via model loss, in self-paced learning (Kumar et al., 2010; Jiang et al., 2015), increasing the contribution of harder training instances over time. This setting posed a challenge due to the fast-changing pace of the loss during training, thus later approaches used human-intuitive difficulty metrics, such as sentence length or the existence of rare words (Platanios et al., 2019) to pre-compute difficulties of training instances. However, as such metrics do not express difficulty of the model, model-based metrics have been proposed over the years, such as measuring the loss difference between two checkpoints (Xu et al., 2020b) or model translation variability (Wang et al., 2019b; Wan et al., 2020). In our curricula we use training dynamics to measure example difficulty, i.e. metrics that consider difficulty from the perspective of a model towards a certain task. Example difficulty can be also estimated either in a static (offline) or dynamic (online) manner, where in the latter training instances are evaluated and re-ordered at certain times during training, while in the former the difficulty of each example remains the same throughout. In our experiments we adopt the first setting and consider static example difficulties.

Transfer Teacher CL is a particular family of such approaches that use an external model (namely the teacher) to measure the difficulty of training examples. Notable works incorporate a simpler model as the teacher (Zhang et al., 2018) or a larger-sized model (Hacohen and Weinshall, 2019), as well as using similar-sized learners trained on different subsets of the training data. These methods have considered as example difficulty, either the teacher model perplexity (Zhou et al., 2020), the norm of a teacher model word embeddings (Liu et al., 2020), the teacher's performance on a certain task (Xu et al., 2020a) or simply regard difficulty as a latent variable in a teacher model (Lalor and Yu, 2020). In the same vein, we also incorporate Transfer Teacher CL via teacher and student models of the same size and type. However, differently, we take into account the behavior of the teacher *during the course of its training* to measure example difficulty instead of considering its performance at the end of training or analysing internal embeddings.

Moving on to **Schedulers**, these can be divided into discrete and continuous. Discrete schedulers, often referred to as *bucketing*, group training instances that share similar difficulties into distinct sets. Different configurations include accumulating buckets over time (Cirik et al., 2016), sampling a subset of data from each bucket (Xu et al., 2020a; Kocmi and Bojar, 2017) or more sophisticated sampling strategies (Zhang et al., 2018). In cases where the number of buckets is not obtained in a straightforward manner, methods either heuristically split examples (Zhang et al., 2018), adopt uniform splits (Xu et al., 2020a) or employ schedulers that are based on a continuous function. A characteristic approach is that of Platanios et al. (2019) where at each training step a monotonically increasing function chooses the amount of training data the model has access to, sorted by increasing difficulty. As we will describe later on, we experiment with two established schedulers and propose modifications of those based on training dynamics.

Other tasks where CL has been employed include Question Answering (Sachan and Xing, 2016), Reading comprehension (Tay et al., 2019) and other general NLU classification tasks (Lalor and Yu, 2020; Xu et al., 2020a). Others have developed modified curricula in order to train models for code-switching (Choudhury et al., 2017), anaphora resolution (Stojanovski and Fraser, 2019), relation extraction (Huang and Du, 2019), dialogue (Saito, 2018; Shen and Feng, 2020) and self-supervised NMT (Ruiter et al., 2020), while more advanced approaches combine it with Reinforcement Learning in a collaborative teacher-student transfer curriculum (Kumar et al., 2019).

3 Methodology

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be a set of training data instances. A curriculum is comprised of two main elements: the *difficulty metric*, responsible for associating a training example to a score that represents a notion of difficulty and the *scheduler* that determines the type and number of available instances at each training step t. We experiment with three difficulty metrics derived from training dynamics and four schedulers: two are new contributions and the remaining are referenced from previous work.

3.1 Difficulty Metrics

As aforementioned, we use training dynamics (Swayamdipta et al., 2020), i.e. statistics originally introduced to analyse dataset quality, as difficulty metrics. The suitability of such statistics to serve as difficulty measures for CL is encapsulated in three core aspects. Firstly, TD are straightforward. They can be easily obtained by training a single model on the target dataset and keeping statistics about its predictions on the training set. Secondly, TD correlate well with model uncertainty and follow a similar trend to human (dis)agreement in terms of data annotation, essentially combining the view of both worlds. Finally, TD manifest a clear pattern of separating instances into distinct areas–*easy*, *ambiguous* and *hard* examples for a model to learn–something that aligns well with the ideas behind Curriculum Learning.

The difficulty of an example $(\mathbf{x}_i, y_i)$ can be determined by a function f, where an example i is considered more difficult than example j if $f(\mathbf{x}_i, y_i) > f(\mathbf{x}_j, y_j)$. We list three difficulty metrics that use statistics during the course of a model's training, as follows:

CONFIDENCE (CONF) of an example $\mathbf{x}_i$ is the average probability assigned to the gold label y_i by a model with parameters θ across a number of epochs E. This is a continuous metric with higher values corresponding to easier examples.

$$f_{\text{CONF}}(x_i, y_i) = \mu_i = \frac{1}{E} \sum_{e=1}^{E} p_{\theta^{(e)}}(y_i | x_i) \quad (1)$$

CORRECTNESS (CORR) is the number of times a model classifies example $\mathbf{x}_i$ correctly across its training. It takes values between 0 and E. Higher correctness indicates easier examples for a model to learn.

$$f_{\text{CORR}}(x_i, y_i) = \sum_{e=1}^{E} o_i^{(e)},$$

$$o_i^{(e)} = \begin{cases} 1 & \text{if } \arg\max p_{\theta^{(e)}}(x_i) = y_i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

VARIABILITY (VAR) of an example $\mathbf{x}_i$ is the standard deviation of the probabilities assigned to the gold label y_i across E epochs. It is a continuous metric with higher values indicating greater uncertainty for a training example.

$$f_{\text{VAR}}(x_i, y_i) = \sqrt{\frac{\sum_{e=1}^{E} \left(p_{\theta^{(e)}}(y_i|x_i) - \mu_i\right)^2}{E}} \quad (3)$$

Confidence and correctness are the primary metrics that we use in our curricula since low and high values correspond to hard and easy examples respectively. On the other hand, variability is used as an auxiliary metric since only high scores clearly represent uncertain examples while low scores offer no important information on their own.

3.2 Schedulers

We consider both discrete and continuous schedulers. Each scheduler is paired with the metric that is most suited, i.e. the discrete correctness with annealing and the continuous confidence with competence.

The **Annealing** (CORR$_{\text{ANNEAL}}$) scheduler proposed by Xu et al. (2020a), assumes that training data are split into buckets $\{d_1 \subset D, \ldots, d_K \subset D\}$ with possibly different sizes $|d_i|$. In particular, we group examples into the same bucket if they have the same *correctness* score (see Equation (2)). In total, this results in $E+1$ buckets, which are sorted in order of increasing difficulty. Training starts with the easiest bucket. We then move on to the next bucket by also randomly selecting $1/(E+1)$ examples from each previous bucket. Following prior work, we train on each bucket for one epoch. The **Competence** (CONF$_{\text{COMP}}$) scheduler was originally proposed by Platanios et al. (2019). Here, we sort examples based on the *confidence* metric (see Equation (1)), and use a monotonically increasing function to obtain the percentage of available training data at each step. The model can use only the top K most confident examples as instructed by this function. A mini-batch is then sampled uniformly from the available examples.

In addition to those schedulers, we introduce the following modifications that take advantage of the *variability* metric. **Correctness + Variability Annealing** (CORR+VAR$_{\text{ANNEAL}}$) is a modification of the Annealing scheduler and **Confidence + Variability Competence** (CONF+VAR$_{\text{COMP}}$) is a modification of the Competence scheduler. In both variations, instead of sampling uniformly across available examples, we give higher probability to instances with high *variability* scores (Equation (3)), essentially using two metrics instead of one. We assume that since the model is more uncertain about such examples further training on them can be beneficial. For all curricula, after the model has finished the curriculum stage, we resume training as normal, i.e. by random sampling of training instances.

3.3 Transfer Teacher Curriculum Learning

In a transfer teacher CL setting a teacher model is used to obtain the difficulty of training examples (Matiisen et al., 2019). As such, the previously presented difficulty metrics are suitable to be used in a transfer teacher CL scenario, since in order to obtain them a teacher model should be fine-tuned on a target dataset.

The two-step procedure that we follow in this study is depicted in Figure 1. Initially a model (the *teacher*) is fine-tuned on a target dataset and training dynamics are collected during the course of training. The collected dynamics are then converted into difficulty metrics, following Equations (1)-(3). In the second stage, the difficulty metrics and the original training data are fed into a scheduler that re-orders the examples according to their difficulty (in our case from easy-to-hard) and feeds them into another model (the *student*) that is the same in size as the teacher.

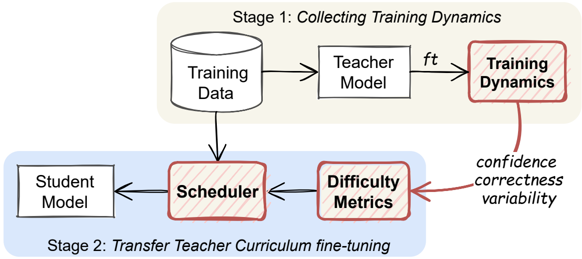


Figure 1: Transfer Teacher Curriculum Learning used in our study. A teacher model determines the difficulty of training examples by collecting training dynamics during fine-tuning (Stage 1). The collected dynamics are converted into difficulty metrics and into a student model via a scheduler (Stage 2).

4 Experimental Setup

4.1 Datasets

In this work we focus on four NLU classifications tasks: Natural Language Inference, Paraphrase Identification, Commonsense Causal Reasoning and Document Classification. The datasets that we use include datasets from the GLUE benchmark: RTE, QNLI and MNLI (Wang et al., 2018) and four cross-lingual datasets: XNLI (Conneau et al., 2018), PAWS-X (Yang et al., 2019), XCOPA (Ponti et al., 2020) and MLDoc (Schwenk and Li, 2018) that combined cover 25 languages. We also use OOD test sets, including NLI Diagnostics (Wang et al., 2018), TwitterPPBD (Lan et al., 2017), CommonSenseQA (Talmor et al., 2019) and HANS (McCoy et al., 2019). The corresponding statistics are shown in Table 1 and more details can be found in Appendix A.

4.2 Evaluation Settings

For models, we use the pre-trained versions of base RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020) from the HuggingFace library[2] (Wolf et al., 2020). For all datasets, we report accuracy as the main evaluation metric across three random

[2] https://huggingface.co/roberta-base, https://huggingface.co/xlm-roberta-base

Train Set	ZS	ID	OOD	Languages	# Train	# Val.	# ZS Test	# ID Test	# OOD Test
PAWS	PAWS-X	PAWS	TwitterPPDB	7	49,401	2,000	2,000	2,000	9,324
MNLI	XNLI	MNLI-m	NLI Diagnostics	15	392,702	2,490	5,010	9,815	1,105
SIQA	XCOPA	SIQA	CSQA	12	33,410	100	500	2,224	1,221
MLDoc	MLDoc	-	-	8	10,000	1,000	4,000	-	-
QNLI	-	QNLI	-	1	99,505	5,238	-	5,463	-
RTE	-	RTE	HANS	1	2,365	125	-	277	30,000

Table 1: Datasets statistics. ZS, ID and OOD correspond to zero-shot Cross-lingual transfer, in-distribution and out-of-distribution settings, respectively. ZS Validation and Test statistics are per language.

Train	PAWS			SIQA		
Test	PAWS (ID)	Twitter (OOD)	*Time* ↓	SIQA (ID)	CSQA (OOD)	*Time* ↓
Random	94.77 ±0.14	72.80 ±5.45		68.36 ±0.39	44.61 ±0.96	
$\text{CR}_{\text{anneal}}$	94.47 ±0.26	72.83 ±6.65	1.00	68.45 ±0.69	44.85 ±0.72	1.00
$\text{Corr}_{\text{anneal}}$	94.72 ±0.09	71.97 ±2.69	0.56 (0.35)	**69.20** ±0.48	45.81 ±1.40	1.28 (1.11)
$\text{Conf}_{\text{comp}}$	94.82 ±0.09	75.18 ±6.71	1.28 (0.72)	67.25 ±1.80	43.93 ±1.59	1.13 (0.57)
$\text{Corr+Var}_{\text{anneal}}$	94.68 ±0.20	72.62 ±1.17	0.77 (0.29)	67.54 ±0.43	44.31 ±0.88	0.71 (0.26)
$\text{Conf+Var}_{\text{comp}}$	**94.88** ±0.14	**81.33** ±2.10	1.20 (0.69)	68.54 ±0.04	**45.84** ±0.67	1.48 (0.71)

Train	MNLI			RTE			QNLI	
Test	MNLI-m (ID)	Diag. (OOD)	*Time* ↓	RTE (ID)	HANS (OOD)	*Time* ↓	QNLI (ID)	*Time* ↓
Random	87.31 ±0.22	61.87 ±1.36		75.57 ±1.19	59.98 ±2.66	-	92.60 ±0.18	-
$\text{CR}_{\text{anneal}}$	87.71 ±0.16	61.78 ±0.27	1.00	74.01 ±2.9	57.26 ±3.18	1.00	92.45 ±0.27	1.00
$\text{Corr}_{\text{anneal}}$	87.53 ±0.23	62.15 ±0.94	0.76 (0.47)	**76.17** ±1.06	55.15 ±2.9	0.76 (0.57)	92.57 ±0.14	1.30 (1.11)
$\text{Conf}_{\text{comp}}$	87.36 ±0.42	61.31 ±1.00	1.33 (0.50)	75.69 ±1.62	55.05 ±1.25	1.11 (0.78)	92.68 ±0.21	1.30 (1.00)
$\text{Corr+Var}_{\text{anneal}}$	87.64 ±0.03	**62.57** ±1.32	1.50 (0.81)	75.45 ±2.23	58.12 ±5.76	1.00 (0.66)	**92.84** ±0.27	1.08 (0.89)
$\text{Conf+Var}_{\text{comp}}$	**87.74** ±0.27	61.82 ±0.98	1.49 (0.60)	76.05 ±1.23	**60.69** ±2.15	1.01 (0.78)	92.63 ±0.13	1.27 (1.07)

Table 2: Accuracy results of RoBERTa on in-distribution (ID) and out-of-distribution (OOD) data. *Time* corresponds to the ratio $S_{*\text{TD}}/S_{\text{CR}_{\text{anneal}}}$, where the numerator is the number steps a curriculum with TD needs to reach the reported performance and the denominator is the number of steps the $\text{CR}_{\text{anneal}}$ baseline requires to reach its performance. Results are reported over 3 random seeds and in parenthesis we include the minimum time required across seeds.

seeds, on the following settings.

ID/OOD: Monolingual models (RoBERTa) are trained and evaluated on English in-distribution and out-of-distribution datasets.

Zero-Shot: Constitutes the zero-shot cross-lingual transfer setting, where a multilingual model (XLM-R) is trained on English data only and tested on languages other than English (Hu et al., 2020).

In all experiments, we select the best checkpoint based on the *English validation set* performance. When reporting significance tests we use the Approximate Randomization test with all seeds (Noreen, 1989). More details about experimental settings can be found in Appendix B.1.

4.3 Model Comparisons

We primarily compare all curricula that use training dynamics against each other and against a baseline (*Random*) that does not employ any curriculum and is using standard random order training. We also consider as another baseline the teacher-transfer curriculum proposed by Xu et al. (2020a), namely *Cross-Review* (indicated as $\text{CR}_{\text{anneal}}$ in the next sections). This curriculum uses the annealing scheduler, but does not employ training dynamics as difficulty scores. Instead, the method splits the training set into subsets and a model is trained on each subset containing $1/N$ of the training set. The resulting models are then used to evaluate all examples belonging in different subsets. The difficulty score of an example is considered the number of its correct classifications across teachers. The difference between the CR metric and the *correctness* metric is that Cross-Review uses N fully trained teacher models on subsets of data, while the latter uses E epochs of a single model trained on the entire training set. We split each training set into 10 subsets for all datasets except MLDoc where we split into 5 and RTE where we split into 3, following prior work.

Finally, when comparing $\text{CR}_{\text{anneal}}$ with our TD curricula, with discrete and continuous schedulers, we ensure that all of them are trained for equal amount of time, resulting in a one-to-one comparison. To enforce this, after the end of the curriculum phase, training continues as normal for the remaining steps by randomly sampling examples.

TRAIN	PAWS		MNLI		SIQA		MLDOC	
TEST	PAWS-X (ZS)	*Time* ↓	XNLI (ZS)	*Time* ↓	XCOPA (ZS)	*Time* ↓	MLDOC (ZS)	*Time* ↓
PRIOR WORK	84.90*	-	75.00*	-	60.72	-	77.66	-
RANDOM	84.49 ±0.08		73.93 ±0.18		60.62 ±0.54		**86.74** ±0.46	
CR$_{\text{ANNEAL}}$	84.35 ±0.46	1.00	74.57 ±0.40	1.00	60.44 ±0.39	1.00	86.59 ±0.29	1.00
CORR$_{\text{ANNEAL}}$	**84.70** ±0.15	1.04 (0.85)	73.92 ±0.11	1.11 (1.09)	60.95 ±0.40	2.13 (0.77)	86.47 ±0.64	1.09 (1.02)
CONF$_{\text{COMP}}$	84.51 ±0.45	1.44 (1.11)	74.32 ±0.41	1.10 (0.53)	61.09 ±0.28	1.33 (0.8)	86.30 ±0.70	1.37 (1.18)
CORR+VAR$_{\text{ANNEAL}}$	84.52 ±0.27	0.75 (0.61)	**74.66** ±0.06	0.79 (0.49)	**61.68** ±0.51	2.73 (1.75)	86.14 ±0.23	0.99 (0.56)
CONF+VAR$_{\text{COMP}}$	84.03 ±0.65	1.50 (1.10)	74.43 ±0.18	1.17 (0.93)	61.04 ±0.31	1.32 (0.58)	85.78 ±0.74	1.20 (0.94)

Table 3: Zero-shot performance between curricula as the average accuracy across languages (mean and standard deviation over 3 random seeds) with XLM-R. We also report prior work results for reference as follows: PAWS-X (Chi et al., 2021), XNLI (Chi et al., 2021), XCOPA (Ponti et al., 2020), MLDoc (Keung et al., 2020) (mBERT). *Note that Chi et al. (2021) tune on the target languages validation sets.

5 Experiments

5.1 Performance & Training Time

Results on Tables 2 and 3 show performance and training time for various datasets. In particular, the reported numbers (*Time*) are calculated as the ratio $S_{*\text{TD}} / S_{\text{CR}_{\text{anneal}}}$, i.e. the number of steps the TD curriculum needs to reach best performance ($S_{*\text{TD}}$) divided by the number of steps the Cross-Review method needs to reach its best performance ($S_{\text{CR}_{\text{anneal}}}$). We focus comparison between curricula to show the tradeback between performance and time. A lower score indicates a larger speedup. In addition, we report in parentheses the minimum time obtained across 3 random seeds.

Table 2 shows accuracies for RoBERTa models when tested on ID or OOD data. We observe that CL has minimal improvements in ID and in particular, through statistical testing we find that the increases over the Random baseline or Cross-Review are not significant for any of the datasets, except for MNLI-M versus Random. Nevertheless, when tested on OOD performance improvement is larger. CONF+VAR$_{\text{COMP}}$ achieves the best performance on TwitterPPDB (+8.5 points, significance $p < 0.01$), CommonSenseQA (+1.23 points) and HANS (+0.71 points, $p < 0.01$ with CR) while CORR+VAR$_{\text{ANNEAL}}$ performs best for NLI Diagnostics (+0.7 points). We speculate that CONF+VAR$_{\text{COMP}}$ achieves higher OOD performance thanks to its slow pacing and the more accurate difficulties of confidence. However, this comes at the cost of speedup by requiring either the same or a few more steps than CR$_{\text{ANNEAL}}$.

Investigating the cross-lingual transfer results on Table 3, initially we observe that CL with XLM-R seems to have a larger impact in terms of performance. On XNLI there is a $+0.73$ points increase over Random ($p < 0.01$). The difference with CR is not significant but TD achieved a 20% speedup on average. On XCOPA we observe $+1.06$ points increase, requiring however more training time with the CORR+VAR$_{\text{ANNEAL}}$ curriculum, over the random baseline. It is worth noting that for XCOPA, the competence-based curricula are able to also offer better performance with less additional training time. As for the remaining datasets, CL is unable to achieve any performance improvement on MLDoc while on PAWS-X CORR$_{\text{ANNEAL}}$ has an improvement of $+0.2$ points from Random and $+0.35$ from CR$_{\text{ANNEAL}}$, both statistically significant, with the cost of no speedup. As another drawback, CR is generally more resource demanding since it needs N fully-trained teacher models instead of 1.

5.2 Comparing Difficulties

We now present a comparison between task-agnostic (TA) and task-specific (TS) difficulty metrics. We re-implement 3 additional difficulty metrics proposed in prior work for Neural Machine Translation. The first two, introduced in Platanios et al. (2019), correspond to sentence length (LENGTH) computed as the number of words in each sentence and word rarity (RARITY) computed as the negated logarithmic sum of the frequency of each word in a sentence. Frequencies are computed over the training set. Finally, we experiment with Perplexity (PPL) as the difficulty of a sentence (Zhou et al., 2020). We calculate sentence perplexity as the average perplexities of its subwords by masking one subword at a time and using the remaining context to predict it. Since we test on a task with two-sentence input, we sum PPL of the two sentences and consider the entire input for LENGTH and RARITY.

TRAIN TEST	PAWS PAWS (ID)	 PAWS-X (ZS)	 TWITTER (OOD)	MNLI MNLI-M (ID)	 XNLI (ZS)	 NLI DIAG. (ODD)
CR$_{\text{ANNEAL}}$	94.47 ±0.26	84.35 ±0.46	72.83 ±6.65	87.71 ±0.16	74.57 ±0.40	61.78 ±0.27
CORR$_{\text{ANNEAL}}$	94.72 ±0.09	**84.70** ±0.15	71.97 ±2.69	87.53 ±0.23	73.92 ±0.11	62.15 ±0.94
CONF$_{\text{COMP}}$	94.82 ±0.09	84.51 ±0.27	75.18 ±6.71	87.36 ±0.42	74.32 ±0.41	61.31 ±1.00
CORR+VAR$_{\text{ANNEAL}}$	94.68 ±0.20	84.52 ±0.27	72.62 ±1.17	87.64 ±0.03	**74.66** ±0.06	**62.57** ±1.32
CONF+VAR$_{\text{COMP}}$	**94.88** ±0.14	84.03 ±0.65	81.33 ±2.10	**87.74** ±0.27	74.43 ±0.18	61.82 ±0.98
LENGTH	94.87 ±0.10	84.56 ±0.09	74.93 ±5.66	87.22 ±0.15	73.47 ±0.29	61.25 ±0.17
RARITY	94.48 ±0.06	84.16 ±0.24	79.90 ±2.70	87.38 ±0.10	73.42 ±0.25	62.25 ±1.08
PPL	94.55 ±0.43	84.09 ±0.30	**83.02** ±1.23	87.27 ±0.10	73.42 ±0.18	61.83 ±0.81

Table 4: Task-specific (above the line) vs Task-agnostic metrics (below the line) on ID, ZS and OOD data.

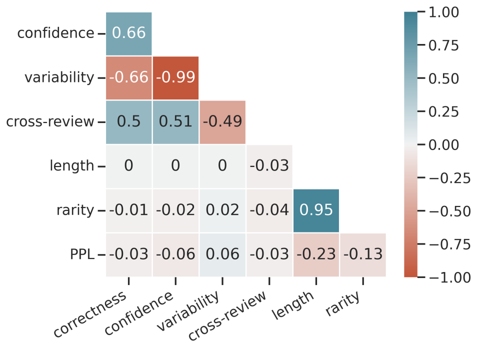


Figure 2: PAWS Spearman rank correlation between difficulty metrics.

Table 4 shows the results of the comparison between metrics on the PAWS and MNLI datasets. Interestingly, we observe that TA metrics perform on par with TS on ID data, worse on ZS data and can perform quite well for OOD data. In particular, RARITY is the third best on Twitter and the second best on NLI Diagnostics. This can be explained by the very different language used on Twitter vs Wikipedia in the training corpus and the human-created data on NLI, which is not that strong in the latter. PPL is the best performing system in Twitter and we find statistically significant improvement ($p < 0.01$) compared with CONF+VAR$_{\text{COMP}}$. Masked word prediction of unknown words could be an informative signal for a very new domain.

Further, we analyse the relation of different difficulty metrics by calculating the Spearman rank correlation between all possible combinations. As shown in Figure 2, we observe very high correlation between confidence and correctness, as expected, but also a good correlation with Cross-Review, explaining their close performance. On the contrary, variability is negatively correlated with those metrics as higher values indicate more uncertainty from the model towards an example. As such, a combination of these opposing metrics can offer benefits than combining two already correlated metrics. Compared with task-agnostic metrics, interestingly, we see almost no correlation with either LENGTH, RARITY or PPL, indicating that examples that the model deems difficult when fine-tuned on a task are very different than those before fine-tuning. RARIRY and LENGTH highly correlate as longer sentences are more likely to contain rare words. Finally, PPL is reverse analogous to them, probably because longer sentences have more context and it is thus easier for the model to predict the masked token. Overall, PPL has a slight positive relation with variability since both measure model uncertainty and high PPL of words might make the model to further fluctuate between its predictions.

5.3 Learning Curves

In order to examine the behavior of the curricula during the course of training, we further plot the average language performance on the validation set as a function of the number of training steps when using XLM-R models for the improved datasets (XNLI and XCOPA). In Figure 3 we draw the best performing curriculum (CONF+VAR$_{\text{COMP}}$), the CR$_{\text{ANNEAL}}$ curriculum and the Random baseline.

A first finding is that for CR$_{\text{ANNEAL}}$ we observe a performance drop around 20K steps in XNLI. Further investigation revealed that the drop happens when the curriculum starts accessing the examples of the last bucket–which is the hardest one. This drop possibly indicates that buckets created by CR do not contain incrementally challenging examples that can help the model prepare for the hardest instances adequately, in contrast with training dynamics that result in smooth training. In addition, we observe that after a point in training (60K) random training stabilises while CONF+VAR$_{\text{COMP}}$ con-

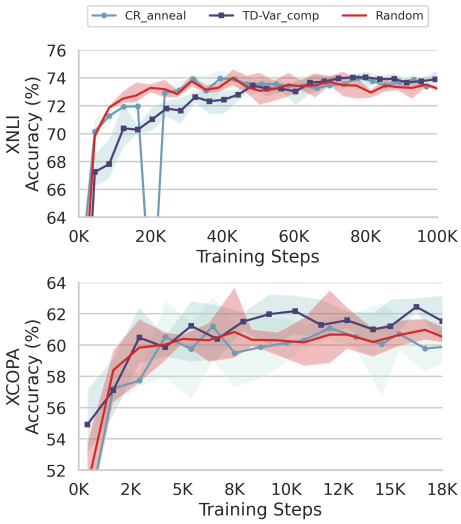


Figure 3: Average validation set accuracy across languages as a function of learning steps (in thousands) with XLM-R models. Results are reported over 3 random seeds.

tinues to improve (70K-120K), despite having an initially lower performance than other schedulers. Regarding XCOPA, the CONF+VAR$_{\text{COMP}}$ curriculum is superior than random training and CR$_{\text{ANNEAL}}$ by consistently improving performance from quite early in training (from step 8K and after).

5.4 Training with limited budget

Since training a teacher model can add overhead to the general training process (training a teacher model plus a similar-sized student), we further conduct a minimal experiment on PAWS, where we collect training dynamics for a teacher XLM-R model for different number of epochs (stopping training early) and then train a student XLM-R model for 10 epochs. Results are reported in Table 5 for our best overall curriculum for this dataset CORR+VAR$_{\text{ANNEAL}}$ as the average of the validation set languages performance.

We observe that it is not necessary to collect training dynamics for a long period of training (e.g. 10 epochs) as even with much less training, for instance 3 epochs, we can still get close performance to prior work much faster. Compared to Cross-Review, that essentially requires full training of N teacher models plus the student model, TD offer a much more efficient solution. Comparing training time with the PPL baseline, TD is even faster as collecting sentence perplexities for the entire PAWS training set requires 1 hour and 30 minutes vs 36 minutes that are needed for 3 epochs of fine-tuning

Teacher Epochs	CR$_{\text{ANNEAL}}$	CORR$_{\text{ANNEAL}}$	*Time* ↓
3		85.20 ± 0.17	0.3
4	85.28 ± 0.18	85.46 ± 0.25	0.4
5		84.94 ± 0.30	0.5
10		85.34 ± 0.19	1.0

Table 5: Validation set performance (average across languages) on PAWS-X with XLM-R models. Student is trained for 10 epochs, while training dynamics are collected from the teacher for different number of epochs.

XLM-R. Ultimately, even having less accurate dynamics (by training the teacher for less epochs) we can achieve overall less training time for the curriculum while still maintaining good performance. Longer teacher training might be proven beneficial for future training of different student versions.

6 Conclusion

We presented a set of experiments using training dynamics (Swayamdipta et al., 2020) as difficulty metrics for CL on several NLU tasks. Differently from existing works, we focus our evaluation on in-distribution, out-of-distribution and zero-shot cross-lingual transfer data by testing existing discrete and continuous schedulers as well as modifications of those in a transfer-teacher curriculum setting.

Our findings offer evidence that simply reordering the training examples in a meaningful way has mostly an impact on zero-shot cross-lingual transfer and OOD data, with no improvement on ID. Our proposed Continuous scheduler with confidence and variability sampling provided a boost up to 8.5% on a challenging OOD dataset over prior work. Comparing our proposed application of training dynamics to other transfer-teacher curriculum methods that are using more than 1 teacher model, we observed greater speedups, improved performance and more stable training. In particular, we found that task-agnostic metrics do not perform better than task-specific ones on ID and ZS data but can offer good performance on OOD settings.

Overall, our experiments suggest there is no curriculum outperforming others by a large margin which is consistent with findings in Zhang et al. (2018) and that task-agnostic metrics should not be rejected when transferring to challenging new domains. However we show that training dynamics are potentially better difficulty metrics for CL in both monolingual and multilingual models even with a limited budget.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. 2021. XLM-E: Cross-lingual language model pre-training via electra. *arXiv preprint arXiv:2106.16138*.

Monojit Choudhury, Kalika Bali, Sunayana Sitaram, and Ashutosh Baheti. 2017. Curriculum design for code-switching: Experiments with language identification and language modeling with deep neural networks. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 65–74, Kolkata, India. NLP Association of India.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. 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 389–398, Hong Kong, China. Association for Computational Linguistics.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2694–2700. AAAI Press.

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.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. 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 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

John P. Lalor and Hong Yu. 2020. Dynamic data selection for curriculum learning via ability estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 545–555, Online. Association for Computational Linguistics.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

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

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. 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 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Dana Ruiter, Josef van Genabith, and Cristina España-Bonet. 2020. Self-induced curriculum learning in self-supervised neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2560–2571, Online. Association for Computational Linguistics.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463, Berlin, Germany. Association for Computational Linguistics.

Atsushi Saito. 2018. Curriculum learning based on reward sparseness for deep reinforcement learning of task completion dialogue management. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 46–51, Brussels, Belgium. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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

Lei Shen and Yang Feng. 2020. CDL: Curriculum dual learning for emotion-controllable response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 556–566, Online. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2019. Improving anaphora resolution in neural machine translation using curriculum learning. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 140–150, Dublin, Ireland. European Association for Machine Translation.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019b. Improving back-translation with uncertainty-based confidence estimation. 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 791–802, Hong Kong, China. Association for Computational Linguistics.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When do curricula work? In *International Conference on Learning Representations*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020a. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages

6095–6104, Online. Association for Computational Linguistics.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020b. Dynamic curriculum learning for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3977–3989, Barcelona, Spain (Online). International Committee on 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.

Mingyang Yi, Lu Hou, Jiacheng Sun, Lifeng Shang, Xin Jiang, Qun Liu, and Zhi-Ming Ma. 2021. Improved ood generalization via adversarial training and pretraining. *arXiv preprint arXiv:2105.11144*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019a. Curriculum learning for domain adaptation in neural machine translation. 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 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. 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.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

A Datasets

In this study, we use the following datasets:

GLUE () is a benchmark for Natural language Understanding tasks. We use a subset of the included datasets: MNLI, RTE and QNLI that are identify textual entailment (3 categories in the first one and 2 for the othet two). Since the test set is hidden can results can be obtained only via submission to the benchmark, we sub-sample a 5% portion from each training set and use it as our validation set. Then, final results are reported on the officially provided validation set.

PAWS-X (Yang et al., 2019) is the cross-lingual version of the English Paraphrase Adversaries from Word Scrambling dataset (Zhang et al., 2019b) containing paraphrase identification pairs from Wikipedia. It consists of human translated pairs in six topologically distinct languages. The training set contains only English examples taken from the original PAWS dataset. As OOD we use the TwitterPPDB dataset (Lan et al., 2017).

XNLI is the cross-lingual NLI dataset (Conneau et al., 2018), an evaluation set created by extending the development and test sets of the MultiNLI dataset (Williams et al., 2018) and translating it into 14 languages. Training data constitutes the original MultiNLI English training set. A OOD we use NLI Diagnostics (Wang et al., 2018), a set of human-annotated examples that reveal model behavior on particular semantic phenomena.

XCOPA is the Cross-lingual Choice of Plausible Alternatives (Ponti et al., 2020), a typologically diverse multilingual dataset for causal common sense reasoning in 11 languages. The dataset consists of development and test examples for each language, which are translations from the English COPA (Roemmele et al., 2011) validation and test sets. Following Ponti et al. (2020) we use the Social IQA dataset (Sap et al., 2019) as training data (containing 3 possible choices), and the English COPA development set as validation data (containing 2 possible choices). For OOD, we consider the CommonSenseQA (CSQA) dataset (Talmor et al., 2019) that contains 5 possible choices.

MLDoc is a document classification dataset with 4 target categories: corporate/industrial, economics, government/social, and markets (Schwenk and Li, 2018). The dataset is an improved version of the Reuters benchmark (Klementiev et al., 2012) consisting of 7 languages and comes with 4 different sets of English training data (1k, 2k, 5k, 10k). Here, we use the 10k following prior work (Keung et al.,

	RoBERTa$_{\text{base}}$	XLM-R$_{base}$
MNLI	7.5 h	11.5 h
PAWS	1.0 h	1.8 h
SIQA	1.0 h	1.3 h
MLDoc	-	1.0 h
QNLI	-	
RTE		

Table 6: Training time required for a full model training.

2020).

Additional datasetss

B Training Details

Hyper-parameter Settings: For all the reported experiments we used the HuggingFace Transformers library with PyTorch[3]. We use base models, XLM-R and RoBERTa with 470M and 340M parameters respectively. We fix sentence length to 128 for all datasets except MLDoc where we use 256. We did minimal learning rate tuning on each dataset's English validation set, searching among [7e-6, 1e-5, 2e-5, 3e-5] and choosing the best performing one (1e-5 for PAWS, 7e-6 for SIQA and MNLI, 3e-5 for MLDoc, 2e-5 for RTE and 2e-5 for QNLI). We clip gradients to 1.0 after each update, use AdamW optimizer (Loshchilov and Hutter, 2017) without any warmup and a batch size of 32 for PAWS, MNLI, QNLI and MLDoc, 8 for SIQA and 16 for RTE. All reported experiments use the same 3 random seeds and all models were trained on a single Nvidia V100 16GB GPU. In terms of training time, Table 6 shows the training time required for each dataset with the above parameters.

Multiple Choice QA: We treat SIQA-XCOPA as a sentence-pair classification task and feed the model a (premise-question, choice) tuple converting each *cause* into “What was the cause?" and each *effect* into “What was the effect?" question which is concatenated to the premise. Similar to prior work (Ponti et al., 2020) we use a feed forward linear layer on top of the input's first special token (<s> in the case of RoBERTa and XLM-R) to produce a score for each of the possible choices. In the case of CSQA that does not have a premise, we simply feed the network the question-choice pair.

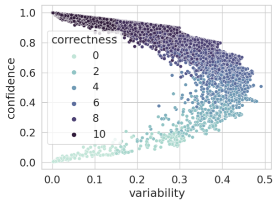
B.1 Curriculum Parameters

In order to collect TD we first fine-tune either a RoBERTa or an XLM-R model on the English training set of each dataset. TD for each example are collected over 10 epochs on MNLI, PAWS and SIQA, while for RTE, QNLI and MLDoc we train for 5 epochs. The COMPETENCE and COMPETENCE VARIABILITY schedulers require to set in advance the number of steps, i.e. total duration of the curriculum phase. We employ the same parameters as in Platanios et al. (2019) and set this value to 90% of steps that the baseline model requires to achieve its best performance on the development set. The initial competence is set to 0.01 for all datasets. We evaluate each model at the end of each epoch and at regular intervals (Dodge et al., 2020), every 500 updates for MNLI (corresponding to 24 times per epoch) and 10 times per epoch for the rest of the datasets. Performance is reported over three random seeds.

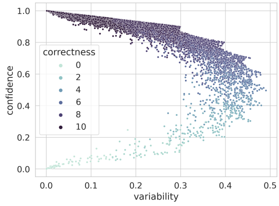
C Analysing Data Maps

Finally, to better understand the reason for the reported CL benefits we plot data maps that result from training an XLM-R model on each dataset in Figure 4, with confidence in the y-axis, variability in the x-axis and correctness in the legend. As observed, the easiest overall datasets, i.e. PAWS-X (4b) and MLDoc (4g) result in quite crisp maps with very few hard-to-learn examples, while in XNLI (4d) and SIQA (4f) the data maps are very dense and the number of difficult examples is high. This can potentially explain why CL with XLM-R models was more beneficial on those datasets in terms of performance, confirming that CL can be used to better prepare a model for harder instances.

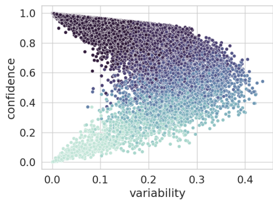
[3] https://pytorch.org/



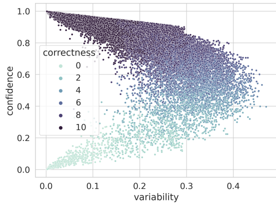
(a) PAWS XLM-R



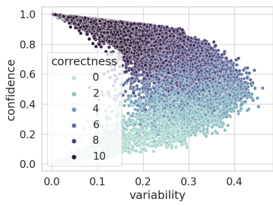
(b) PAWS RoBERTa



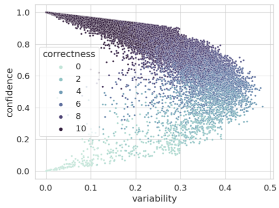
(c) MNLI XLM-R



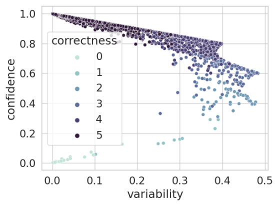
(d) MNLI RoBERTa



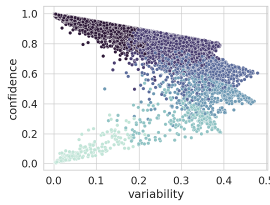
(e) SIQA XLM-R



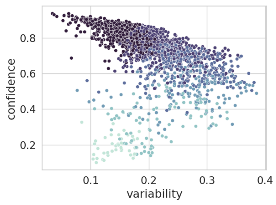
(f) SIQA RoBERTa



(g) MLDoc XLM-R



(h) QNLI RoBERTa



(i) RTE RoBERTa

Figure 4: Data map for the training set of each dataset. We plot maximum 25K examples for clarity.