Modeling Easiness for Training Transformers with Curriculum Learning

Leonardo Ranaldi [*,•], **Giulia Pucci**[*], **Fabio Massimo Zanzotto**[*]
(•) Idiap Research Institute, Martigny, Switzerland
[*] Department of Enterprise Engineering, University of Rome Tor Vergata, Rome, Italy
[first name].[last name]@uniroma2.it

Abstract

Directly learning from complex examples is generally problematic for humans and machines. Indeed, a better strategy is exposing learners to examples in a reasonable, pedagogically-motivated order. Curriculum Learning (CL) has been proposed to import this strategy when training machine learning models. In this paper, building on Curriculum Learning, we propose a novel, linguistically motivated measure to determine example complexity for organizing examples during learning. Our complexity measure - LRC- is based on length, rarity, and comprehensibility. Our resulting learning model is CL-LRC, that is, CL with LRC. Experiments on downstream tasks show that CL-LRC outperforms existing CL and non-CL methods for training BERT and RoBERTa from scratch. Furthermore, we analyzed different measures, including perplexity, loss, and learning curve of different models pre-trained from scratch, showing that CL-LRC performs better than the state-of-the-art.

1 Introduction

Pre-trained Transformers are sweeping away all other methods of natural language understanding. These models outperform all previous methods and sometimes even humans in many NLP tasks (Wang et al., 2018, 2020; Kalyan et al., 2021; Guo et al., 2022). Pre-training on unlabeled large-scale corpora seems to be the way that increases performance (Ranaldi et al., 2022). For example, BERT is pre-trained on an English corpus of 3.300 million words consisting of books (Zhu et al., 2015) and Wikipedia. However, training these models with large corpora is quite expensive in terms of computation time and memory.

The problem of optimizing the computational resources that Transformers need is tackled in three main ways: by re-modeling pre-training tasks (Yang et al., 2019a; Clark et al., 2020), by studying techniques to produce lighter architectures (Sanh et al., 2019; Liu et al., 2019), and by working with data (Moore and Lewis, 2010; Gururangan et al., 2020; Chang et al., 2021).

Architecture-level and model-level optimization techniques have been extensively studied in the context of pre-training methods for NLP. Data-level approaches have yet to be explored. To this end, we adopted a data-level strategy called Curriculum Learning (CL), which stems from the complexity of training samples so that the model can achieve better performances.

Starting from the idea for which humans and animals acquire first elemental concepts and then, gradually, more complex ones, Bengio et al. (2009) proposed CL and demonstrated its benefits in shape recognition. This approach presents training data in order of difficulty, starting with easy examples and increasing the degree in parallel with learning.

The application of CL in Pre-trained Language Models (PLMs) has limitations. One of the most critical challenges is to find a criterion for measuring the difficulty of training samples. In supervised tasks, sorting training batches by length and repetitiveness of certain patterns paid off (Kocmi and Bojar, 2017; Chang et al., 2021). In the semi-supervised PLMs, word representations are learned by optimizing loss in the masked language modeling tasks using a set of contiguous blocks of fixed-length text. Nagatsuka et al. (2021) proposed a CL strategy focused on training the self-attention mechanism from shorter blocks to longer ones. This is because each head of this mechanism seems to be more attentive to local dependencies than global ones (Kovaleva et al., 2019; Sukhbaatar et al., 2019; Podkorytov et al., 2021).

In this paper, building on Curriculum Learning, we propose a novel, linguistically motivated measure to determine example complexity. This measure - LRC- is based on length, rarity, and compre-

Proceedings of Recent Advances in Natural Language Processing, pages 937–948
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_101

hensibility and sorts text complexity into blocks that increase in dimensionality gradually during pre-training of BERT and its variants.

Moreover, by exploiting the organization of the example, our method avoids the loss of context common in standard CL methods applied to PLMs (Nagatsuka et al., 2021). Using a small-scale corpus, experimental results demonstrated that our approach outperforms the other methods on GLUE tasks, and it requires fewer examples to achieve the same results. Finally, we showed that CL-LRC achieves sustainable performance compared to CL in terms of perplexity, loss, and learning curves of the different models pre-trained from scratch.

2 Related Works

The main studies for optimizing computational resources and increasing the learning capabilities of Pre-trained Language Models (PLMs) are architecture-based, learning model-based, and, finally, data-driven. Although previous works have demonstrated the functionality of architecture-level and model-level approaches, they still need to improve. Yang et al. (2019b), have introduced permutation language modeling that allows models to capture bidirectional contexts and has performed well on long-dependency contexts but requires more data and computational resources to train and deploy. Clark et al. (2020), have reduced computational costs by modifying the traditional MLM with a discriminator that, in turn, could have limitations in tasks that require a deep understanding of long-term dependencies or complex relationships between words and concepts. Sanh et al. (2019); Lan et al. (2020) have used parameter reduction techniques and have achieved a light version of BERT that is faster and more lightweight but is not as effective as BERT in tuning parameters on specific tasks. Liu et al. (2019) have improved BERT pre-training by introducing dynamic masking in the MLM task and eliminating the NSP task. These structural changes are the key to increasing the model's performance in downstream tasks, but more data are needed to achieve the same results than in the pre-training of BERT. The performance achieved by optimization at the architecture and training levels is a difficult point of resistance to overcome. While these topics have been extensively studied in the context of PLMs, the data-level approach still needs to be explored.

Although numerous variants of BERT succeed in fixing some critical aspects of pre-training, there open up many gaps at the computational and performance level on downstream tasks. Many studies have found that the multi-headed self-attention mechanism requires more computational effort. Since each head of this mechanism seems to be more attentive to local dependencies than global ones (Kovaleva et al., 2019; Sukhbaatar et al., 2019; Podkorytov et al., 2021), training local self-attention in shorter blocks seems to be less complex than training global self-attention in longer blocks. Therefore, using the size of the input text block is key to measuring the difficulty level of the training samples. For these reasons, Nagatsuka et al. (2021) have proposed a Curriculum Learning (CL) strategy focused on hands-on training of the self-attention mechanism. In particular, they applied the strategy directly to BERT pre-training, exploiting the input text block size in the context of the self-attention mechanism as a measure of difficulty for BERT pre-training.

Beyond the world of PLMs, many studies on CL have used sentence length, external resources, or input sequences to measure difficulty in various NLP tasks. Spitkovsky et al. (2010) have proposed a CL-based method for parsing tasks. Kocmi and Bojar (2017) have proposed a text length-based method on no transformer-based models for tasks of neural machine translation. While Xu et al. (2020) also included the rarity of some terms by applying the method for the reading comprehension task. Lee et al. (2022) propose a gradual masking mechanism of concepts for pre-training the language model that obtains impressive results but is tied to the knowledge graph. In this paper, we propose text complexity techniques coupled with input text block size in the context of the self-attention mechanism. The two approaches are used to measure the difficulty of BERT pre-training. Our proposal adds a further light step where pre-training text complexity is computed to the incremental CL proposed in (Nagatsuka et al., 2021). Our model achieves higher performance than other methods on downstream tasks.

3 Methods

Since language has a structure, organizing examples during pre-training can improve model performance. Curriculum Learning (CL) is a training method based on the idea that training algorithms can achieve better results when training data are

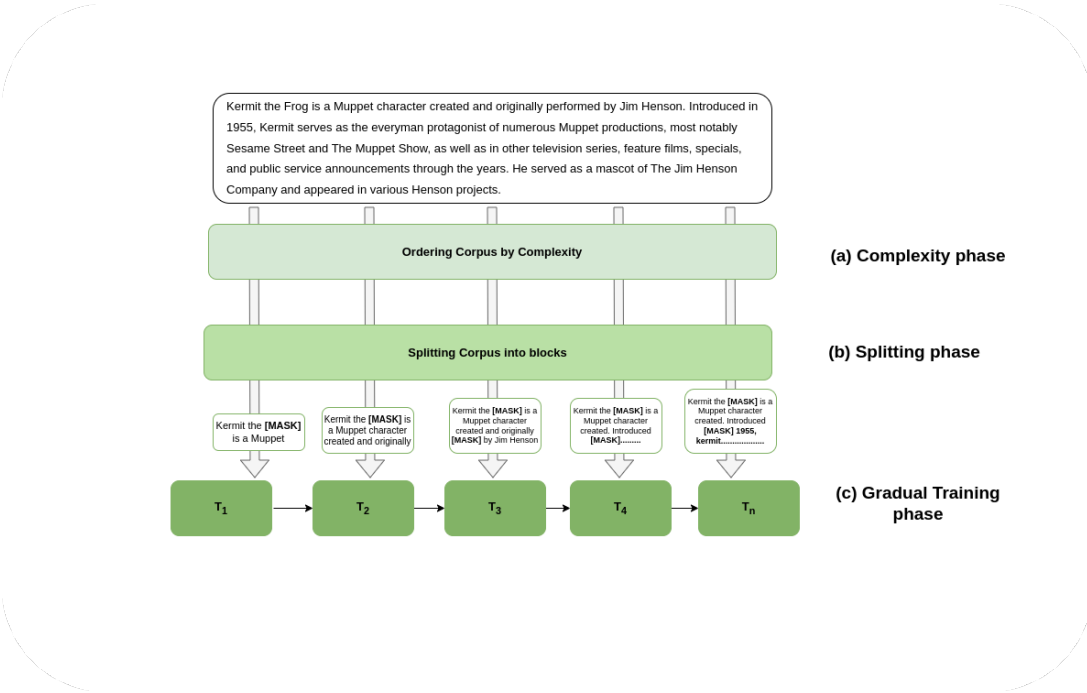

Figure 1: Curriculum Learning method overview.

presented in accordance with the model's current skills. We propose CL-LRC that adds to the standard CL, a measure used to determine example complexity during the pre-training (see Figure 1).

The CL-LRC method consists of three phases: (a) sorting the corpus according to our complexity measure, (b) partitioning the corpus according to specific block sizes, and (c) gradual pre-training by increasing block sizes. Firstly, we sorted the corpus by complexity measure, starting with the less complex sentences to more complex ones (Section 3.2). Secondly, we split the sorted corpus into a series of input blocks of predefined length (Section 3.3.2). Finally, we trained a model by shifting the training samples from the short block-size to the long one, depending on the predefined number of training steps (Section 3.3.3). Pre-training was done by masking some block tokens randomly, as precedes the Masked Language Modeling (MLM) task (Devlin et al., 2019). In this section, we describe the MLM task and the details of the three phases of our CL-LRC approach.

3.1 Masked Language Modeling

BERT training consists of two phases: pre-training and fine-tuning. Two semi-supervised tasks are performed during pre-training: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Liu et al. (2019) in RoBERTa eliminated the NSP task by showing that it did not have a significant benefit on the model's overall performance in downstream tasks and may even have a negative impact on performance, as it introduces noise and bias into the model. For this reason, in this paper, we focus only on MLM by making a methodology adaptable for both BERT and RoBERTa.

During MLM, tokens in a block are randomly masked. About 15% of the tokens are masked (Devlin et al., 2019), and the model is asked to predict the original tokens. It allows processing a bidirectional context without information leaking between layers. Given the sequence $s = w_0, w_2, ..., w_T$ of tokens, where T is the block size. Randomly masking an arbitrary number of tokens, an input sequence $\hat{s}$ is obtained. Given the corrupted sequence $\hat{s}$, MLM predicts the original sequence s. The training objective is formulated as:

$$\max_{\theta} \log p_\theta(s|\hat{s}) \approx \sum_{i=0}^{T} m_i \log p_\theta \left(w_i|w_{<i}, w_{>i},\right) \quad (1)$$

where w_i is the expected token at the position, and i and θ are the model parameters. m_i is a flag indicating the presence of a masked token. If w_i is masked $m_i = 1$, otherwise 0.

Sentence	$d_L(s)$	$d_R(s)$	$d_C(s)$	$d_{LRC}(s)$
= = Major themes = = The Feast of the Goat's major themes include political corruption, machismo, memory, and writing and power. Olga Lorenzo, reviewer for The Melbourne Age, suggests that overall Vargas Llosa's aim is to reveal the irrational forces of Latin tradition that give rise to despotism.	45	0.17	10.5	0.33
= = Reign = = According to the Augustan History, Odaenathus was declared king of Palmyra as soon as the news of the Roman defeat at Edessa reached the city. It is not known if Odaenathus contacted Fulvius Macrianus and there is no evidence that he took orders from him.	46	0.17	10.3	0.41

Table 1: Examples of the complexity values produced by the metrics defined in section 3.2.

3.2 Complexity

Our complexity measure - LRC - is the core of our method. The complexity of a textual example is reflected in many ways, e.g., the length of the context, the use of rare words, or the magnitude of the learning goal. Since the Masked Language Modeling task should aim to learn language from context merely as humans do, these heuristics seem fitting for the Curriculum Learning of PLMs. Firstly, we used the sentence length heuristic to compute the length of sentences of the pre-training corpus (3.2.1). Secondly, we used the rarity heuristic to compute the rarity of words in the corpus (3.2.2). Finally, we used the comprehensibility metric or, more commonly, Flesch-Kincaid readability (3.2.3). The aggregation of these three values forms d_{LRC}, the cornerstone element of our model (3.3.1).

In the rest of this section, we denote our training corpus as a collection of D sentences, $\{s_i\}_{i=0}^{D}$, where each sentence is a sequence of words denoted with $s_i = \{w_0^i, w_1^i, ..., w_n^i\}$.

3.2.1 Sentence Length

Complexity is built on sentence length, starting from the intuition that longer sequences are more difficult to encode and that there may be a likelihood that they will be cut off, thus losing context (Kocmi and Bojar, 2017). Therefore, longer sentences would be more prone to the loss of context in MLM. Although Devlin et al. (2019) are not concerned about this problem, the work proposed by Nagatsuka et al. (2021) uses different truncations shorter than the value recommended in (Liu et al., 2019). It is defined as:

$$d_L(s_i) = length(s_i) \tag{2}$$

we calculate this value for each sentence s_i of our corpus D, obtaining the $d_{L_{max}}$ and $d_{L_{min}}$, which are the maximum and minimum values of the lengths. Finally, we normalize the values:

$$\hat{d}_L(s_i) = \frac{d_L(s_i) - d_{L_{min}}}{d_{L_{max}} - d_{L_{min}}}, \forall i \in [0, |D|]. \tag{3}$$

3.2.2 Rarity

The rarity of words in a sentence, introduced by Platanios et al. (2019), is defined as the probability product of unigrams. This metric implicitly represents information about the sentence length since the scores of longer sentences are the sum of more words and thus are likely to be more significant. Given a corpus of sentences, $\{s_i\}_{i=0}^{D}$, the complexity metric for word rarity is defined as:

$$d_R(s_i) \triangleq -\sum_{k=1}^{N_i} \log p\left(w_k^i\right) \tag{4}$$

where we use logarithms of word probabilities to prevent numerical errors. Note that negation is used because we define less likely (i.e., rare) sentences as more complex. The component $p(w)$ is defined as:

$$p(w) \triangleq \frac{1}{N_{total}} \sum_{i=1}^{M} \sum_{k=1}^{N_i} \mathbb{1}_{w_k^i = w} \tag{5}$$

for each w unique word in corpus and $\mathbb{1}_{condition}$ is the indicator function which is equal to 1 if its condition is satisfied and 0 otherwise. We calculate this value for each sentence s_i of our corpus D, obtaining the $d_{R_{max}}$ and $d_{R_{min}}$, which are the maximum and minimum rarities for sentences. Finally, we normalize the values:

$$\hat{d}_R(s_i) = \frac{d_R(s_i) - d_{R_{min}}}{d_{R_{max}} - d_{R_{min}}}, \forall i \in [0, |D|]. \tag{6}$$

3.2.3 Readability Metric

Common factors for measuring comprehensibility or more common readability are Speed of perception, Perceivability in peripheral vision, Reflex blink technique, Speed Reading, Eye movements, Reading fatigue, Cognitively motivated features, Word difficulty, and N-gram analysis. Unfortunately, it is not always possible to capture all these features.

Accordingly, we used the Flesch-Kincaid metric (Talburt, 1986). This metric is a tool used to assess the comprehensibility of a text. It is based on the length of sentences and words within a text and provides a score that indicates the text's difficulty level. The lower the score, the easier it is to read and comprehend the text. The formula for calculating the Flesch-Kincaid Grade Level score is as follows:

$$d_C(s_i) = 0.39\frac{avg(d_L(s_i))}{100} + 11.8\frac{avg(d_L(w_i))}{100} - 15.59 \quad (7)$$

where $avg(d_L(s_i))$ average sentence length is the number of words in a sentence divided by the number of sentences, and $avg(d_L(w_i)$ is the average word length, i.e. is the number of syllables per word divided by the number of words. The value 0.39 is used to scale the effect of the average sentence length so that it can be compared to the effect of the average word length, weighted by the value 11.8. The final score is then adjusted by subtracting the value of 15.59, which is used to adjust the score scale to match the grading levels used in education more closely. We calculate this value for each sentence s_i and obtain the maximum $d_{C_{max}}$ and the minimum $d_{C_{min}}$ scores. Finally, we normalize these values:

$$\hat{d}_C(s_i) = \frac{d_C(s_i) - d_{C_{min}}}{d_{C_{max}} - d_{C_{min}}}, \forall i \in [0, |D|]. \quad (8)$$

3.3 Curriculum Learning with LRC

This section describes how we utilize the above complexity metrics in the Curriculum Learning approach.

3.3.1 Applying Complexity Heuristics

In the first phase, we estimate the complexity of each sentence $d_{LRC}(s_i)$ by adding the normalized values of length $\hat{d}_L(s_i)$, rarity $\hat{d}_R(s_i)$, and readability score $\hat{d}_C(s_i)$, that is:

$$d_{LRC}(s_i) = \hat{d}_L(s_i) + \hat{d}_R(s_i) + \hat{d}_C(s_i) \quad (9)$$

Then, we sort the sentences of the original corpus by order of increasing complexity before the pre-training phase. Finally, we recompose the re-ordered corpus ready for pre-training. Table 1 shows the values for three examples from the WikiText-2 corpus sorted by their respective complexity values. These heuristics are lightweight, using only 16GB of memory, we can process up to 20k sentences per second for calculating sentence rarity scores and up to 150k sentences per second for calculating sentence length scores.

3.3.2 Splitting a Corpus-Based on Block-sizes

In the second phase, following the directions of Nagatsuka et al. (2021), we divided the original corpus into training samples of the specified size. Each input text for BERT pre-training, called 'block' (Devlin et al., 2019), should not be linguistically consistent as a sentence but a fixed interval of contiguous text. Thus, it is not guaranteed either that the input is a period or that it begins with the first word of a sentence. Moreover, after extensive experiments, Liu et al. (2019) argue that it is desirable for the input sequence to be at most 512 tokens. So we follow this approach to obtain the block of a given length from the corpus as a training sample. The difference is the order, which is the reason why it could be easier for a transformer to learn by order of complexity. We trained a Byte-Pair Encoding (BPE) at the byte level (Radford et al., 2019) to split the raw text into a sequence of tokens. Byte-level BPE allows the decomposition of words, including words outside the vocabulary likely to appear during testing, especially when using a small training dataset. In the experiment, we set the vocabulary size to $20,000$.

3.3.3 Gradual Training

In the third phase, we trained a step-by-step model with four different block sizes, namely 64, 128, 256, and 512, using the corpus sorted by complexity order. At first, we trained the model with the shortest block size, 64, for an arbitrary number of steps. Then, we retrained the model with block sizes of 128 and 256, respectively, for the same number of steps. Finally, we retrained the model with the most extended block size of 512 until it converges. We masked the 15% of tokens as recommended in (Devlin et al., 2019). When restarting training, we continuously initialized the learning rate. We used the maximum batch size available based on the block size to speed up training, as

	Natural Language Inference				Similarity & Paraphrase			Single Sentence
Model	**WNLI**	**RTE**	**QNLI**	**MNLI**	**QQP**	**MRPC**	**SST-2**	**CoLA**
Baseline (BERT)	57.73	52.16	59.63	55.63	68.41	69.85	80.56	**72.40**
Baseline (RoBERTa)	56.83	52.26	64.13	58.43	69.81	69.45	79.22	64.50
Total-Curriculum (BERT)	56.71	52.98	75.93	**67.36**	75.69	74.43	83.35	68.77
Total-Curriculum (RoBERTa)	56.83	53.42	78.71	66.18	76.35	72.79	83.48	**65.72**
Anti-Curriculum (BERT)	55.46	50.67	53.67	58.12	69.87	64.26	78.94	69.74
Anti-Curriculum (RoBERTa)	56.83	52.34	49.46	60.64	72.88	70.09	80.38	62.86
$Curriculum_{LRC}$ *(BERT)*	**60.88**	**58.12**	**79.22**	66.49	**81.16**	**76.11**	**87.16**	71.26
$Curriculum_{LRC}$ *(RoBERTa)*	**57.28**	**56.05**	**81.13**	**66.25**	**78.68**	**74.26**	**85.94**	65.19
$Anti - Curriculum_{LRC}$ *(BERT)*	56.44	50.33	54.32	57.95	69.12	65.11	79.21	69.16
$Anti - Curriculum_{LRC}$ *(RoBERTa)*	57.04	51.95	49.67	61.13	72.45	70.43	80.46	62.23

Table 2: Table of accuracies on GLUE task (Wang et al., 2020).

done in (Nagatsuka et al., 2021).

4 Experimental Results and Discussion

In the experiments, we evaluated our proposed CL-LRC approach in model performance. Therefore, we show that performances increase to the proposed state of the art in (Nagatsuka et al., 2021). In order to reproduce the results proposed in previous work, we used Wikitext-2 (Merity et al., 2017) for pre-training BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). For fine-tuning downstream tasks, we used the famous General Language Understanding Evaluation (GLUE) dataset (Wang et al., 2018). This choice was made to have terms of comparison with state of the art and for ease of retrieval in the huggingface library (Wolf et al., 2019). Finally, we performed an ablation study, perplexity, loss, and learning curves on different subsets of the dataset. All experiments were performed on two NVIDIA RTX A6000 with 48 GB of memory. The code and model will be released for further research.

4.1 Data

Pre-training: BERT and RoBERTa are commonly trained with large corpora, i.e., bookcorpus and Wikipedia-dump with about 3 billion words (Zhu et al., 2015). In this work, we used Wikitext-2 (Merity et al., 2017), a small corpus for simulations, allowing pre-training with a limited computational resource. Wikitext-2 is a standard language model corpus with 720 good-quality articles from English Wikipedia.

Fine-tuning: We fine-tuned the previously introduced models on GLUE benchmarks (Wang et al., 2018). GLUE consists of eight tasks to measure the generalization performance of pre-trained language models. The tasks in question are SST-2, MRPC, QQP, MNLI, QNLI, RTE, WNLI, and CoLA.

4.2 Experimental setup

We performed three methods: the baseline, the Total-Curriculum, a CL proposed by Nagatsuka et al. (2021), and our CL-LRC named $Curriculum_{LRC}$. Hence, we conducted the experiments proposed in (Nagatsuka et al., 2021) using RoBERTa to observe CL on different architectures, and we also reproduced the experiments with BERT. Close to the baseline and Total-Curriculum of BERT and RoBERTa, respectively, we developed our proposed CL-LRC, $Curriculum_{LRC}$, consisting of three steps. First, we sorted the corpus according to complexity, as introduced in section 3.2. Second, we sorted the corpus according to the training samples' difficulty level, using the training samples' block-size as a metric, as explained in section 3.3.2. Finally, we performed the stepwise pre-training phase by increasing the block size defined in section 3.3.3.

We used BERT and RoBERTa, which have 12 layers with a hidden size of 768, where each layer has 12 attention heads. In addition, we used AdamW with a learning rate of 1e-5 in pre-training with four different batch sizes based on the block sizes. In the various proposed training, the models were trained for $10,000$ steps with each block dimension, except for the last block dimension, where training continued until the models converged. For a comparative evaluation, we trained BERT and RoBERTa without CL, using random sampling as the base model with the block dimensionality set to 512, as recommended in (Devlin et al., 2019). Finally, in fine-tuning, we employed the same optimizer used in pre-training, and we set a learning rate of 5e-5 and a batch-size of 64 for all tasks. The total CL time is given by the training time for each training step corresponding to each block dimension.

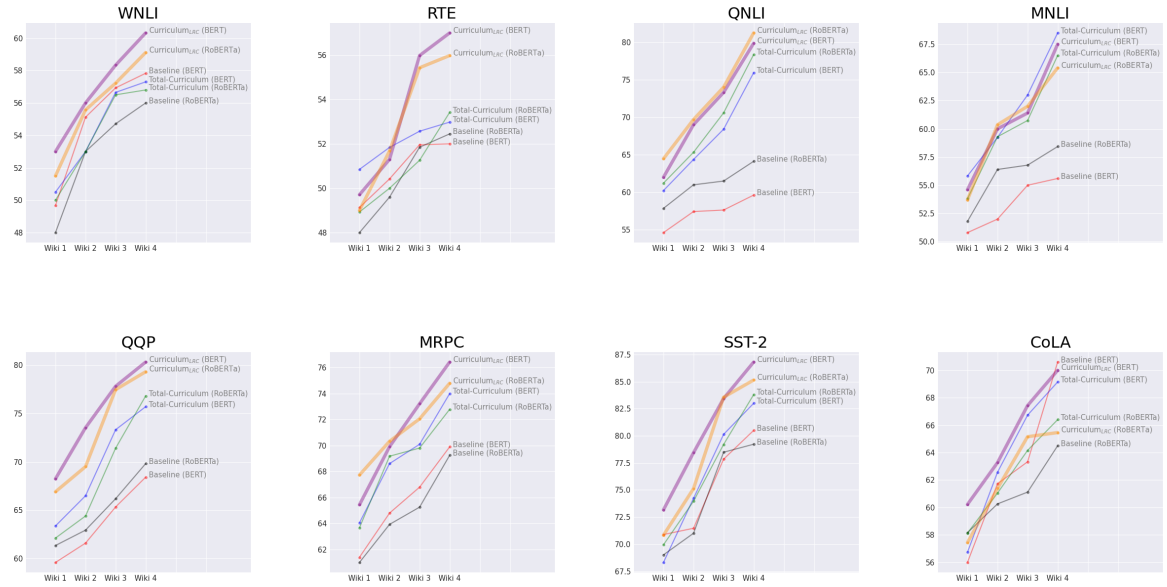

Figure 2: Curriculum Learning increasing pre-training size.

4.3 Results

The results from linguistically motivated pre-training from the complexity of our CL-LRC, in Tab. 2, Tab. 3 and Fig. 4 named $Curriculum_{LRC}$, outperform models based on standard pre-training and Total-Curriculum proposed in (Nagatsuka et al., 2021). However, the batch-size increase supports the performance achieved by Curriculum Learning. Finally, in Figure 2, learning curves on accuracies explain the trade-off between pre-training corpus size and accuracy. These conclusions are derived from the intrinsic evaluations (perplexity and loss) and the extrinsic evaluations (downstream classification tasks).

4.3.1 Our Methods vs Baseline & Curriculum Learning

In particular, for 6 of 8 downstream tasks, our $Curriculum_{LRC}$ outperformed the baselines and the Total-Curriculum proposed by Nagatsuka et al. (2021). Although the accuracies of the proposed models are low compared to those of Liu et al. (2019) and Devlin et al. (2019) due to small-scale pre-training, improvements can be observed.

Firstly, the performance on WNLI, RTE, QNLI, QQP, MRPC, and SST-2 was superior to the baseline by a wide margin in particular (+17 on QNLI, +8,8 on QQP, +4,8 in MRPC and +6,7 on SST-2) for RoBERTa and (+5,9 on RTE, +19,5 on QNLI, +12,7 on QQP, +6,2 in MRPC and +6,6 on SST-2) for BERT. At the same time, the accuracy of MNLI and CoLa was low in both the curriculum and the baseline.

Secondly, comparing the performance of our $Curriculum_{LRC}$ with the Total-Curriculum proposed in (Nagatsuka et al., 2021), there were considerable improvements (+3.7 on RTE, +2.4 on QNLI, +2.3 on QQP and +2.5 on SST-2) for RoBERTa and (+7.4 on RTE, +3,3 on QNLI, +5,5 on QQP and +4.4 on SST-2) for BERT.

Different from what was achieved in previous tasks in MNLI and CoLA, there were no significant improvements. In MNLI, although there were improvements over baselines, $Curriculum_{LRC}$ does not perform as well as Total-Curriculum for the BERT model; instead, for RoBERTa, our $Curriculum_{LRC}$ outperforms Total-Curriculum and baseline.

In CoLA, although $Curriculum_{LRC}$ outperformed Total-Curriculum, the baselines were higher for the BERT model.

4.3.2 Anti-Curriculum vs Curriculum

In the proposed pre-training, we perform standard-curriculum training where we increase the block-size of the training samples from the shortest to the longest. Similarly, we propose Anti-Curriculum training where the training samples with the longest block size are first given to the model as the most difficult. The difficulty level of the training samples

is gradually reduced by shortening the block size in the training process. By comparing standard-curriculum training with Anti-Curriculum, which follows the opposite sampling order, we show that increasing block-size is an effective CL method for PLMs.

Compared with standard-curriculum models, the performances of the Anti-Curriculum models in Table 2) were lower in all downstream tasks; Nagatsuka et al. (2021) had already observed this phenomenon in RoBERTa, and we confirmed it in BERT as well. Moreover, the effect of the additional level of complexity, which we have named $Anti-Curriculum_{LRC}$, does not contribute, and the performances do not change dramatically. This twofold result shows that increasing block complexity is an effective CL method for PLMs.

4.3.3 Increasing pre-training size

Moreover, we show the learning curve by showing the performance growth trend based on the pre-training corpus size. Hence, we tested the proposed models on different subsets of the pre-training dataset. We considered four Wikitext-2 combinations composed of the 25%, 50%, 75%, and finally, 100% of the original corpus introduced in Section 4.1. We named the sub-portions, respectively, Wiki1-4 concerning the portions considered. Our $Curriculum_{LRC}$ performed well on small portions of the corpus, confirming what was obtained in Table 2. In particular, in the bold lines (Figure 2), it can be seen that our models almost always exceed the baselines. Therefore there is a trend toward increasing the amount of data. By using half of the dataset, our strategy $Curriculum_{LRC}$ reaches the same performance as other methods that use all the datasets, indicating that the structure council, although simple, can empower the model (Zanzotto et al., 2020).

4.3.4 Language Model Pre-training

Finally, we studied training loss and perplexity. Cross-entropy loss and perplexity, defined as the exponentiation of cross-entropy loss, where cross-entropy loss is defined as the negative sum of the mean log-likelihood of LM, are used to measure the model's confidence in the observed sequence.

From the results obtained in Figure 3, we can remark that $Curriculum_{LRC}$ outperforms the baselines of both BERT and RoBERTa in terms of loss during the different training steps. Likewise, more promising results can be seen with a constant trend than Total-Curriculum. Furthermore, from the perplexity as the number of tokens increases, our $Curriculum_{LRC}$ performs better than Baseline and Total-Curriculum for both BERT and RoBERTa. Table 4 confirms the results analyzed during the training, where the final loss and perplexity on the evaluation set are shown.

4.4 Ablation Study

In this section, we delve into our method by studying different complexity heuristics. Hence, close to $Curriculum_{LRC}$, we tested the previously proposed model using the three complexity heuristics in the following way: $Curriculum_L$, $Curriculum_R$, $Curriculum_C$ are composed respectively of $\hat{d}_L(s_i)$, $\hat{d}_R(s_i)$ and $\hat{d}_C(s_i)$, $Curriculum_{LR}$ is composed of the sum of $\hat{d}_L(s_i)$ and $\hat{d}_R(s_i)$, $Curriculum_{RC}$ is composed of the sum of $\hat{d}_R(s_i)$ and $\hat{d}_C(s_i)$, and finally, $Curriculum_{LC}$ is composed of the sum of $\hat{d}_L(s_i)$ and $\hat{d}_C(s_i)$, where $i \in [0, |D|]$.

Downstream of these experiments, we can observe that prevalently aggregation of length, rarity, and comprehensibility outperform other configurations. In five out of eight tasks (see Table 5) $Curriculum_{LRC}$ model achieved the best accuracies. In the remaining tasks, the best results were obtained by $Curriculum_{LR}$ for MRPC and QNLI but only for RoBERTa. In difference, in MNLI, the best result was obtained by the $Curriculum_{RC}$ model. While for the non-aggregated models, i.e., $Curriculum_L$, $Curriculum_R$, $Curriculum_C$, we can observe low downstream performances.

5 Conclusion

In this paper, building on Curriculum Learning, we propose a novel measure, - LRC -, to determine example complexity. This measure is applied during pre-training to sort the corpus according to complexity. Experiments conducted in a low-resource environment have shown that the proposed method leads to better performance in downstream tasks and may be used to reduce the data needed for reasonable performances. Furthermore, this approach is straightforward and thus easy to implement.

In further research, we will expand the corpus and validate the scalability of our approach. In addition, it is important to continue investigating different complexity metrics that could be modified during pre-training and their impact on model performance.

Limitation

The limitations of this study are as follows: The proposed method was evaluated in a low-resource environment, specifically using the Wikitext-2 dataset (Merity et al., 2017). Further experiments on more massive datasets are needed to validate the scalability of the proposed approach. The complexity metric used in this study was based on the length of the input text block. While this metric was sufficient for the scope of this study, it is essential to investigate different complexity metrics and their effects on model performance in future works. This study focused on BERT and RoBERTa models, but it would be beneficial to explore the applicability of the proposed method to other transformer-based models in future research. In summary, the proposed method has been shown to be effective in improving performance on downstream tasks within a limited simulation environment. Future research should focus on further evaluating the scalability of this approach in larger datasets, investigating different complexity metrics, and testing the method with other transformer-based models. Additionally, evaluating the effectiveness of the proposed method in the fine-tuning stage is an interesting direction to pursue.

Acknoledgements

This paper has been supported by Social Tourism E -Platform (STEP) project funded by LAZIOINNOVA research and innovation program under grant agreement No. 35561, DTC TEI1 II Phase.

We would also like to thank all the anonymous reviewers that helped to strengthen this paper.

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.

Ernie Chang, Hui-Syuan Yeh, and Vera Demberg. 2021. Does the order of training samples matter? improving neural data-to-text generation with curriculum learning. In *EACL*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

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.

Shangwei Guo, Chunlong Xie, Jiwei Li, Lingjuan Lyu, and Tianwei Zhang. 2022. Threats to pre-trained language models: Survey and taxonomy.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus : A survey of transformer-based pretrained models in natural language processing.

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.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Mingyu Lee, Jun-Hyung Park, Junho Kim, Kang-Min Kim, and SangKeun Lee. 2022. Efficient pre-training of masked language model via concept-based curriculum masking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7417–7427, Abu Dhabi, United Arab Emirates. 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*, abs/1907.11692.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 989–996, Held Online. INCOMA Ltd.

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.

Maksim Podkorytov, Daniel Biś, and Xiuwen Liu. 2021. How can the [mask] know? the sources and limitations of knowledge in bert. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Leonardo Ranaldi, Aria Nourbakhsh, Arianna Patrizi, Elena Sofia Ruzzetti, Dario Onorati, Francesca Fallucchi, and Fabio Massimo Zanzotto. 2022. The dark side of the language: Pre-trained transformers in the darknet.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759, Los Angeles, California. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, Florence, Italy. Association for Computational Linguistics.

John Talburt. 1986. The flesch index: An easily programmable readability analysis algorithm. In *Proceedings of the 4th Annual International Conference on Systems Documentation*, SIGDOC '85, page 114–122, New York, NY, USA. Association for Computing Machinery.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.0.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019a. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Curran Associates Inc., Red Hook, NY, USA.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–267, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

A Appendix

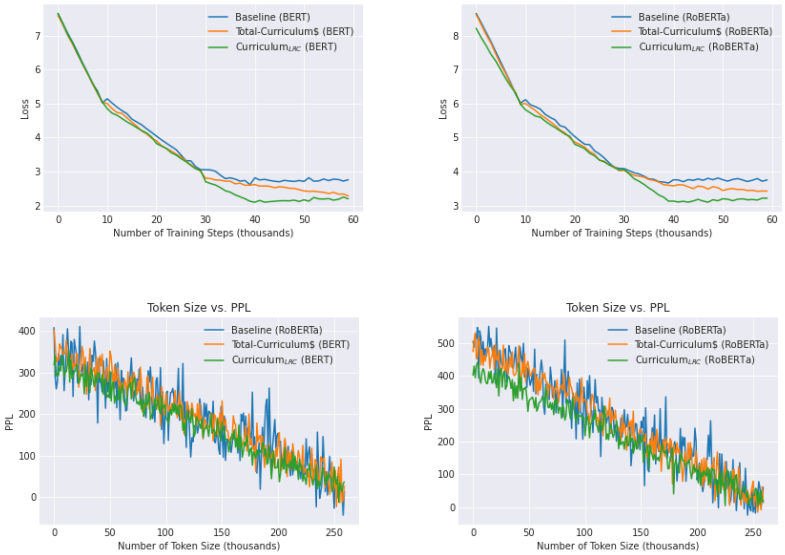

Table 3: Loss and Perplexity during the training phase.

B Appendix B

Model	Loss	Perplexity
Baseline (BERT)	2.7456	15.2844
Baseline (RoBERTa)	2.5122	14.5547
Total-Curriculum (BERT)	2.5678	14.7566
Total-Curriculum (RoBERTa)	2.4172	13.7893
Anti-Curriculum (BERT)	3.2971	16.4327
Anti-Curriculum (RoBERTa)	2.9226	15.2753
$Anti-Curriculum_{LRC}$ *(BERT)*	2.4876	13.6791
$Anti-Curriculum_{LRC}$ *(RoBERTa)*	2.4973	14.5781
$Curriculum_{LRC}$ *(BERT)*	2.2677	12.3356
$Curriculum_{LRC}$ *(RoBERTa)*	**2.1784**	**13.6418**

Table 4: Loss and Perplexity after Pre-training on Evaluation set.

C Appendix

	Natural Language Inference				Similarity & Paraphrase			Single Sentence
Model	**WNLI**	**RTE**	**QNLI**	**MNLI**	**QQP**	**MRPC**	**SST-2**	**CoLA**
$Curriculum_L$ *(BERT)*	57.33	53.16	77.25	65.92	77.54	74.53	82.61	68.92
$Curriculum_L$ *(RoBERTa)*	56.91	53.44	77.23	65.14	77.21	72.95	83.18	63.72
$Curriculum_R$ *(BERT)*	57.28	53.12	77.15	65.82	77.66	74.31	82.66	69.02
$Curriculum_R$ *(RoBERTa)*	56.77	53.61	77.16	65.19	75.11	72.16	83.31	63.69
$Curriculum_C$ *(BERT)*	56.23	52.63	76.25	65.62	76.83	74.41	82.11	68.16
$Curriculum_C$ *(RoBERTa)*	56.22	54.13	76.91	64.12	74.83	71.91	83.19	63.66
$Curriculum_{LR}$ *(BERT)*	60.32	57.26	**79.95**	66.22	80.24	**76.43**	86.81	70.82
$Curriculum_{LR}$ *(RoBERTa)*	57.11	55.68	80.23	65.94	78.53	**74.98**	85.15	64.91
$Curriculum_{RC}$ *(BERT)*	57.94	53.36	77.35	**67.88**	76.12	74.22	82.93	70.82
$Curriculum_{RC}$ *(RoBERTa)*	55.82	53.21	77.41	65.91	75.89	73.06	82.78	65.46
$Curriculum_{LC}$ *(BERT)*	58.22	53.37	78.18	66.72	76.81	74.63	82.96	69.21
$Curriculum_{LC}$ *(RoBERTa)*	55.43	54.17	77.19	**66.87**	76.11	73.28	82.88	64.86
$Curriculum_{LRC}$ *(BERT)*	**60.88**	**58.12**	79.22	66.49	**81.16**	76.11	**87.16**	**71.26**
$Curriculum_{LRC}$ *(RoBERTa)*	**57.28**	**56.05**	**81.13**	66.25	**78.68**	74.26	**85.94**	**65.19**

Table 5: Table of accuracies of our Curriculum Learning method on different complexity measures.