On the Role of Corpus Ordering in Language Modeling

Ameeta Agrawal, Suresh Singh, Lauren Schneider, Michael Samuels
Department of Computer Science
Portland State University, USA
{ameeta, singhsp, lbraun, msamuels}@pdx.edu

Abstract

Language models pretrained on vast corpora of unstructured text using self-supervised learning framework are used in numerous natural language understanding and generation tasks. Many studies show that language acquisition in humans follows a rather structured simple-to-complex pattern and guided by this intuition, curriculum learning, which enables training of computational models in a meaningful order, such as processing easy samples before hard ones, has been shown to potentially reduce training time. The question remains whether curriculum learning can benefit pretraining of language models. In this work, we perform comprehensive experiments involving multiple curricula strategies varying the criteria for complexity and the training schedules. Empirical results of training transformer language models on English corpus and evaluating it intrinsically as well as after fine-tuning across eight tasks from the GLUE benchmark, show consistent improvement gains over conventional vanilla training. Interestingly, in our experiments, when evaluated on one epoch, the best model following a document-level hard-to-easy curriculum, outperforms the vanilla model by 1.7 points (average GLUE score) and it takes the vanilla model twice as many training steps to reach comparable performance.

1 Introduction

Pretrained language models are the foundation for achieving impressive results on many natural language processing tasks today (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019) but they are also prohibitively expensive to train, requiring enormous computing resources to be effective. The exploding demand of computations, together with the resulting massive energy cost (Strubell et al., 2019), pose serious obstacles to the development of new pretrained models, thus, leading to a number of recent research efforts aimed towards addressing this problem by proposing approaches for improving model efficiency (Sanh et al., 2019) or sample efficiency (Clark et al., 2020), to name just a few.

The primary method researchers have explored to address this problem is to develop smaller language models (Sanh et al., 2019; Jiao et al., 2020). In this paper we study a complementary approach to simplify pretraining of language models based on corpus ordering via curriculum learning (Bengio et al., 2009). The motivation is that curriculum learning has been shown to help with faster convergence (Guo et al., 2018; Hacohen and Weinshall, 2019) which, in combination with simpler language models, will give us a sustainable platform for future research into language models.

Although curriculum learning strategies have been successfully employed in many areas of machine learning, on a wide range of tasks (Wang et al., 2020; Soviany et al., 2021), little is understood about the role of corpus ordering in the context of pretraining language models with the exception of some notable early works on language modeling (Bengio et al., 2009; Graves et al., 2017). In this paper, we continue this line of investigation by asking the question whether curriculum based pretraining of *transformer language models* provides any benefits when compared with traditional vanilla training.

In order to create a curriculum from an unlabeled text corpus for such self-supervised form of learning, one needs to define a measure of complexity. We explore if metrics of text readability difficulty designed for humans can serve as beneficial metric in creating a curriculum for machine learning models. The intuition behind this approach is to mimic the manner in which humans learn. Training samples are organized by levels of difficulty and training proceeds in steps where the model is first trained on a subset of the corpus at a given difficulty level before being trained on another difficulty level, and so on.

Proceedings of the 2nd Workshop on Simple and Efficient Natural Language Processing, pages 142–154
November 10, 2021. ©201 Association for Computational Linguistics

For our experiments, we pretrain from scratch multiple models of BERT$_{\text{BASE}}$ using the WikiText-103 corpus (Merity et al., 2016), a collection of English articles from Wikipedia. We explored standard length-based metrics of complexity as well as document-level complexity using Flesch Reading Ease readability index to organize the corpora into a curriculum. We also investigated two different methods of accessing such an ordered training set – one where the bins remain disjoint (BINNED) and another where the bins are incrementally joined together to increase the training data size (STEPPED) over n epochs. An extensive evaluation is conducted in terms of metrics related to not just pretraining but also fine-tuning on eight downstream tasks, specifically the suite of datasets from the GLUE benchmark (Wang et al., 2018).

Our results demonstrate that while sentence-level and document-level complexity metrics work comparably well, they outperform the vanilla models in all experiments. Furthermore, while easy to hard is a good strategy when complexity is computed at sentence-level, the reverse is true when complexity is measured at document-level. Finally, the most encouraging results suggest that corpus ordering takes significantly less time (measured in terms of training steps) as compared to vanilla training, while yielding comparable results as measured by the average GLUE score.

In summary, this work makes the following contributions,

- We propose a novel paradigm defined by document-level metrics of text complexity for ordering a training corpus.
- We explore two strategies for learning from such a curriculum in the context of pretraining of language models.
- We conduct extensive experiments by training several versions of transformer model from scratch and evaluating their performance in terms of metrics computed at both pretraining and fine-tuning stages.

2 Corpus Complexity

We begin by describing the unlabeled corpus of text used during training and the metrics adopted for measuring its complexity.

Corpus. In this work we choose the WikiText-103 corpus (Merity et al., 2016), which has served as a popular choice of text corpus for language modeling in prior works (Khandelwal et al., 2019; Press et al., 2021). It consists of a set of verified Good and Featured articles[1] from English Wikipedia, containing a total of around 103.2 million tokens[2]. For all experiments, the predefined splits of the corpus are used.

Complexity. Before we can order the unlabeled text instances in WikiText-103, we need to define some criteria for ranking them. Designing a ranking criteria in a self-supervised setup is a challenging problem, especially due to the dual pretraining and fine-tuning paradigm. We choose to estimate sample complexity in terms of human-centered notion of ‘difficulty’, specifically text readability and lexical richness. In other words, we construct a heuristically informed pre-defined curriculum, and adopt metrics that measure the difficulty of text, based on readability and complexity, at two levels of granularity - sentence and document. The corpus is then divided into bins, from which batches of training input are constructed at random.

2.1 Sentence-level complexity

The length of a sentence (or a sequence) is a straightforward, intuitive, and attractive (easy to implement) measure for indicating difficulty or complexity of a piece of text, and indeed, numerous works have used length as a criteria for ordering samples (Spitkovsky et al., 2010; Cirik et al., 2016; Kocmi and Bojar, 2017), although, to our knowledge, none in the context of pretraining of *transformer* language models. Note that sentence length as discussed here is different from sequence length (Press et al., 2021), with the former being the number of contiguous words in an entire sentence (linguistic unit), whereas the latter is the number of tokens in a subsequence that serves as an input to the model (machine unit)[3].

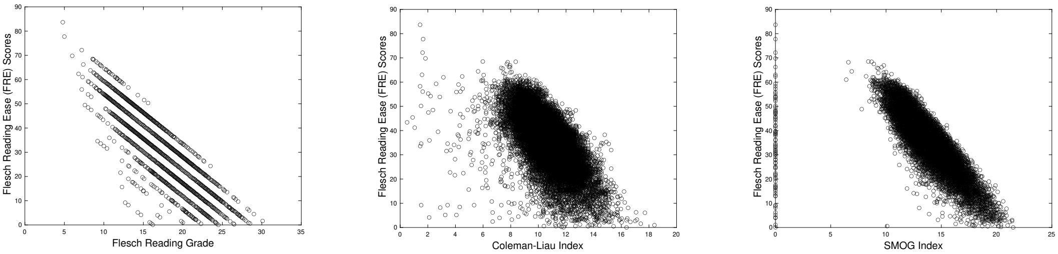
We first split each article into non-overlapping sentences, which are further tokenized into words[4].

[1] In this paper, the words ‘articles’ and ‘documents’ are used interchangeably.

[2] https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/

[3] To further illustrate the difference between the two notions of length, sentence vs. subsequence, consider this very simple sentence ‘*This is a very long sentence.*’. Here, the length of the sentence is 6, whereas subsequences of varying lengths can be created from this sentence, with their length being anywhere between 1 and 6, depending on the user specified parameter. Moreover, the subsequences can be non-overlapping or otherwise (Press et al., 2021).

[4] We use NLTK’s sentence segmentation and word tokenization.



(a) FRE vs Grade (b) FRE vs Coleman-Liau (c) FRE vs SMOG index

Figure 1: Comparison of the four metrics of readability on the WikiText103 corpus.

Length	FRE	Grade	Coleman-Liau	SMOG
2-10	52.8	8.4	8.0	9.2
11-15	49.2	9.8	9.9	10.5
16-20	44.7	11.5	10.3	11.6
21-25	48.5	12.1	10.8	12.6
26-30	43.7	13.9	10.7	13.5
31-35	38.8	15.8	10.9	14.4
36-40	33.9	17.7	10.9	15.2
41-45	29.0	19.6	11.1	16.0
46-50	24.0	21.5	11.1	16.7
51-55	18.9	23.5	11.1	17.4
56-60	05.3	26.6	11.0	18.1
61+	-12.7	33.6	11.2	20.2

Table 1: Sentence length and its connections to Flesch Reading Ease (FRE), Flesch-Kincaid Grade, Coleman-Liau index and SMOG index. For FRE, higher scores reflect easier samples (in terms of readability difficulty); for all others, a smaller value reflects easier to read. In general, there appears to be a consistent monotonic mapping between length and all other metrics, except for an occasional outlier (underlined).

Then we compute the number of tokens in each sentence and create bins consisting of sentences of similar lengths – 2 to 5 words, 6 to 10 words, and so on, together, thus organizing the corpus to create a length-based curriculum.

It is important to note that by binning sentences based on length this way, it is possible that we may (completely) lose the surrounding context of the sentences. However, in looking closely at the bins, we noticed that multiple sentences of similar lengths from the *same* document ended up assigned to the same bin successively, thereby still maintaining some relevant context. Some examples are included in the Appendix A.

In order to address any issues that may arise due to a corpus shuffled so thoroughly, we also explore document-level metrics of complexity.

2.2 Document-level complexity

This time, we conduct a very different analysis of the WikiText-103 focusing on text readability for approximating text complexity. We initially studied four measures of readability[5] to classify the documents – the Flesch Reading Ease (FRE) (Kincaid et al., 1975), Flesch-Kincaid Grade Level (Kincaid et al., 1975), Coleman-Liau Index (Coleman and Liau, 1975), and Smog Index (Mc Laughlin, 1969).

Comparison of the four readability metrics. Figure 1 plots the FRE scores against the other three readability metrics for WikiText computed at the document level. We observe that the FRE score and the Grade-level scores exhibit a linear relationship implying that both these metrics rate the documents similarly. However, there is some variability because at any Grade level we have a range of FRE scores. This is consistent with the manner in which Grade levels are computed since each Grade level is based on a range of computed scores. There is a similar linear relationship between FRE and SMOG index. The relationship between FRE and Coleman-Liau is similar though there is a larger spread of values for a given FRE value. However, we generally observe a linear relationship between FRE values and the other indices, and for this reason we settle on the FRE measure for scoring documents. Further, since FRE has a larger range compared to the other indices, it gives us the fidelity, i.e., fine-grained classification, needed to bin documents.

In FRE, higher scores denote *easier* samples, in terms of their readability difficulty, whereas lower values indicate *harder* samples. Concretely, for a document, FRE is computed as,

$$\text{FRE} = 206.835 - 1.015\left(\frac{\#w}{\#s}\right) - 84.6\left(\frac{\#l}{\#w}\right)$$

where w denotes the words, s denotes sentences, and l denotes syllables.

We also ran the FRE metric on each of the sentence length bins as described earlier. As Table 1

[5] These metrics were computed via TextStat library: https://pypi.org/project/textstat/

Bin	% tokens
easy	32.8%
medium	32.6%
hard	34.5%

Table 2: Statistics of binning the WikiText103 corpus based on FRE scores.

shows, there is indeed a direct correlation between FRE scores and the length of sentences in the bin. As the sentence length increases, FRE scores drop (indicating greater difficulty). To validate this we ran the other three metrics as well and those results are also displayed in the table. Like FRE, the other three metrics also provide strong correlation between sentence length and difficulty.

In order to create a curriculum based on differing levels of complexity, we use a histogram of FRE scores to identify boundaries for splitting the corpus. Additional details on the distribution fit of the FRE sccores are included in Appendix B. Table 2 provides a summary of how we split the corpus into three bins of roughly same size in terms of number of tokens but varying difficulty of documents, namely, *easy*, *medium*, and *hard*. Based on this split, we can consider different curriculum learning strategies including training easy-to-hard or hard-to-easy with variations where either we train on each bin individually or cumulatively, as described in the next section.

As an additional experiment, we also computed the type token ratio score (TTR) of each document, a ratio of the number of unique words (types) to the total number of words (tokens) in a document – the closer the TTR is to 1, the greater the lexical richness, and thus, considered as 'complex' for our purposes. This is because none of the other metrics described earlier take word types into account, which have been shown to be helpful in prior works (Zhang et al., 2018). Appendix C includes some comparison between FRE and TTR scores.

3 Corpus Ordering for Pretraining

We first summarize the process of creating the curriculum and then explain the strategy for training over it. As described in the previous section, sentences or documents are sorted by their complexity score, which are then distributed into non-overlapping bins, essentially subsets of data (also known as shards (Zhang et al., 2018)), such that samples in each bin are similar in complexity.

The training consists of t sequential phases, where t denotes the different points of time during the training, where training samples are fetched only from a subset of bins. For instance, $t = 1$ may correspond to the first epoch or first n steps, $t = 2$ may correspond to the second epoch or the next n steps, and so on. In our experiments, t denotes an epoch. A subset consists of one or more bins, and for creating and iterating over these training subsets during training, we explore two different strategies - BINNED and STEPPED.

(i) **BINNED**: In this variant, the model is trained sequentially on each bin, one at a time. In other words, the model is first trained on the first bin and its state is saved. The training then continues from the saved checkpoint on the next bin, and so on. This is similar to the case where a subset consists of only one bin, and we iterate over it for one epoch.

The bins themselves can be accessed in order of either increasing difficulty (from easy to medium to hard), an approach that can be intuitively seen as mimicking the way humans learn, or in the reverse order of decreasing difficulty (from hard to medium to easy), a technique shown to benefit machine learning algorithms (Weinshall et al., 2018). In doing so, the question we ask is whether curriculum or anti-curriculum help in the context of language modeling, if at all.

It is worth mentioning that while the bins are accessed in a pre-defined order (i.e., easy to hard or reverse), the samples within the bins are still randomly selected, thus combining a deterministic schedule with the benefits of randomization that serve neural models well.

(ii) **STEPPED**: Alternatively, the bins could be accessed cumulatively where the training set progressively increases in size by addition of newer bins while retaining earlier bins (also referred to as Baby Steps curriculum (Bengio et al., 2009; Spitkovsky et al., 2010)). Basically, samples of increasing (or decreasing) complexity are added to the training set after each phase t while the samples from the previous phase still remain in the training set. For instance, at the first phase $t = 1$, only the first bin is presented; at the next phase $t = 2$, the training set comprises of both the first and second bins; at $t = 3$, samples from the first, second, and third bins are accessed, and so on until the model iterates over the entire corpus in the last phase. Essentially, the first bin's samples are seen n more times ($n =$

number of bins) than the last bin. So in our case of three bins, the easy samples are iterated over thrice as compared to the samples from the hard bin if following an increasing level of complexity, and vice versa for a schedule of decreasing complexity.

To sum it up, if we set a phase t to be an epoch, in the BINNED models, all the bins (and by extension, the entire corpus) are/is iterated over just once, whereas in the STEPPED models, since newer bins are progressively added in addition to existing bins in the training set, a process that not only modifies complexity but also increases data size over time, on average, this is similar to iterating over the entire corpus n^2 times.

Finally, the VANILLA model, where the corpus is not sorted or binned in any way and data from the entire corpus is accessed randomly at any phase, serves as the baseline model.

4 Experiments and Discussion

In this section, we perform a series of experiments for evaluating the effects of corpus ordering strategies in the pretraining of language models.

4.1 Evaluation Metrics

We conduct experiments for evaluating different aspects of the language model and report the results pertaining to, (i) **pretraining**: As intrinsic metrics of evaluation, we measure *loss* and *perplexity* on the validation set which provide some insight into the language modeling capabilities of the pretrained models; (ii) **fine-tuning**: The benefits of better pretraining gained in terms of validation loss or perplexity will be lost if they do not transfer over to downstream natural language processing tasks after fine-tuning, which currently remains the predominant way of using these pretrained language representations. Thus, we further fine-tune the pretrained models on a range of task datasets and report metrics of extrinsic evaluation including *F1 scores*, *Spearman Correlations* and *Accuracy*; and, (iii) **compute resources**: Finally, we perform an analysis of the training time measured in the *number of training steps* taken to reach a certain threshold of performance.

4.2 Model Combinations

We train from scratch and fine-tune the following models:

- Training can go from easy to hard (EASY) or hard to easy (HARD).

- The criteria for determining complexity of text is specified at (a) sentence-level using length (LENGTH), or (b) document-level derived from FRE scores (FRE) or TTR scores (TTR).

- For BINNED, the training set consists of any one bin during a phase, whereas for STEPPED learning, the bins are cumulatively added to the training set at each phase.

- Finally, the standard way of training where the original corpus is used as is, without any ordered curriculum, serves as our baseline denoted as the VANILLA model.

4.3 Pretraining

For training all the models from scratch, we adopt a transformer model, i.e., $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), uncased version, with 12 transformer layers, and masked language modeling objective. The batch size is set to 8, the maximum length of the input sequence is 512, and all other settings set as default. First, we train LENGTH and BINNED models for one epoch over each bin (in other words, as the bins are disjoint, each sample in the entire corpus is seen just once). As a comparable baseline, VANILLA is also trained for one epoch.

For the STEPPED models, since the earlier bins are iterated over more times than the latter bins, it has the complexity of O(n^2). Thus we re-run the VANILLA and BINNED models for two epochs for a comparable evaluation.

4.4 Finetuning

Subsequently, we fine-tune each pretrained model for two epochs over eight distinct task datasets from the General Language Understanding Evaluation (GLUE) benchmark, a suite of diverse natural language understanding tasks, including CoLA, MNLI, MRPC, QNLI, QQP, RTE, SST-2 and STS-B, described in detail in Wang et al. (2018).

4.5 Implementation

All runs are trained on a single GPU (Nvidia GeForce RTX 2080) and all pretraining and fine-tuning experiments are performed using HuggingFace[6] library (Wolf et al., 2019).

[6] https://github.com/huggingface/transformers

		Pretraining ↓		Fine-tuning ↑								
Model		Loss	PPL	CoLa	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
VANILLA		**6.09**	**441.2**	69.0	62.0	69.6	60.8	76.9	55.2	79.4	25.1	62.2
BINNED$_{\text{LEN}}$	INC.	6.61	748.5	**69.1**	**63.8**	**72.0**	**61.3**	**80.9**	**55.6**	79.7	25.6	**63.5**
	DEC.	6.72	835.1	**69.1**	62.6	68.8	60.8	78.1	**55.6**	**80.2**	**25.8**	62.6

Table 3: Results of bins derived from sentence length (increasing/decreasing) and vanilla training, in terms of loss and perplexity (↓ is better) on the valid set at the end of training, and F1 scores for QQP and MRPC, Spearman Correlations for STS-B, and accuracy scores for the other tasks after fine-tuning (↑ is better). The last column reports the average GLUE score. Each bin (or the entire corpus in the case of VANILLA model) is iterated over for one epoch.

		Pretraining ↓		Fine-tuning ↑								
Model		Loss	PPL	CoLa	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
VANILLA		6.09	441.2	69.0	62.0	69.6	60.8	76.9	55.2	79.4	25.1	62.2
BINNED$_{\text{LEN}}$	INC.	6.61	748.5	**69.1**	63.8	72.0	61.3	80.9	55.6	79.7	**25.6**	63.5
BINNED$_{\text{TTR}}$	HARD	**5.96**	**388.9**	68.1	31.8	71.5	50.5	81.3	50.9	81.4	23.8	57.4
BINNED$_{\text{FRE}}$	EASY	6.06	430.8	**69.1**	61.9	70.3	**62.1**	**82.2**	55.6	**81.5**	24.5	63.4
	HARD	6.11	450.8	**69.1**	**64.6**	**72.3**	61.4	81.7	**56.0**	81.4	24.5	**63.9**

Table 4: Results of training over bins derived from document level complexity (TTR and FRE scores) in easy to hard (EASY)or hard to easy (HARD) order. Results of vanilla training and length (increasing) are included from Table 3 as baselines. Loss and perplexity (↓ is better) is measured on the valid set at the end of training, and F1 scores for QQP and MRPC, Spearman Correlations for STS-B, and accuracy scores for the other tasks are reported after fine-tuning (↑ is better). The last column reports the average GLUE score. Each bin (or the entire corpus in the case of VANILLA model) is iterated over for one epoch.

4.6 Results

The goal of this work is to examine the role of corpus ordering inspired by curriculum learning in the pretraining of transformer language models. In particular, we seek to explore the following questions:

Q1. Effect of Sentence Complexity (BINNED)

We first compare the simple sentence-level LENGTH based curriculum learning with the VANILLA way of training language models, the results of which are presented in Table 3. We observe that while the VANILLA model performs better in terms of loss and perplexity on the validation set at the end of one epoch, those improvements fail to carry over to the fine-tuning stage. Instead, both the BINNED$_{\text{LEN}}$ curriculum models perform better in terms of the average GLUE score, with the increasing LEN schedule outperforming the other two.

These results are encouraging considering that a simple approach of iterating over the corpus in bins of progressively increasing sentence length seems to be quite effective.

Q2. Effect of Document Complexity (BINNED)

Next we investigate the effect of accessing the corpus organized based on document-level notion of complexity as measured using FRE scores (and some preliminary experiments using TTR scores, which did not perform as well as FRE scores and, therefore, were not included in further experiments), as compared to VANILLA and BINNED$_{\text{LEN (INC.)}}$ models. The results are reported in Table 4 and once again we notice a disconnect between the intrinsic and extrinsic evaluation scores. That is, while BINNED$_{\text{TTR}}$ (HARD) does well in terms of validation loss and perplexity, it is the poorest performing model when measured in terms of average GLUE score after fine-tuning, and instead, the overall best performance is achieved by the BINNED$_{\text{FRE}}$ (HARD) model.

There appears to be little difference between the performance of LENGTH models and FRE models even though it has been noted that it may be better to use a document-level corpus rather than a shuffled sentence-level corpus (Devlin et al., 2019). We hypothesize that this may be because in length-based bins, multiple sentences of similar lengths

		Pretraining ↓		Fine-tuning ↑								
Model		Loss	PPL	CoLa	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
VANILLA		5.98	398.6	**68.7**	**63.5**	73.5	61.2	79.1	55.2	**81.3**	**24.9**	63.4
BINNED$_{FRE}$	EASY	5.97	394.4	68.1	63.2	**73.8**	50.5	77.0	52.7	80.6	**24.9**	61.4
	HARD	6.00	406.5	68.5	62.4	71.8	**62.5**	82.7	57.4	81.2	23.9	**63.8**
STEPPED$_{FRE}$	EASY	**5.86**	**351.7**	67.3	62.7	71.1	50.5	**83.0**	**58.4**	80.2	24.4	62.2
	HARD	5.98	398.2	68.1	63.4	71.3	61.6	82.7	53.4	79.5	24.7	63.1

Table 5: Results of training over bins derived from FRE scores in easy to hard (EASY)or hard to easy (HARD) order using BINNED or STEPPED algorithm. Loss and perplexity (↓ is better) is measured on the valid set at the end of training, and F1 scores for QQP and MRPC, Spearman Correlations for STS-B, and accuracy scores for the other tasks are reported after fine-tuning (↑ is better). The last column reports the average GLUE score. The entire corpus in the case of VANILLA model is iterated over for two epochs whereas for BINNED versions, each bin is iterated over twice.

belonging to the *same* document get binned successively, thereby continuing to maintain their context when long contiguous sequences as extracted for input. The main observation, however, is that some notion of ordering certainly seems to help as compared to no ordering such as in the case of the VANILLA model.

Crucially, this experiment also allows us to analyze the question whether metrics of text readability designed for humans could also benefit machine learning algorithms, and it appears that an anti-curriculum derived from FRE scores, i.e., from *hard to easy* ordering works better for machines. This result is especially interesting and in line with prior work (Weinshall et al., 2018) where it was noted, in the context of visual object recognition, that what makes an image difficult to a neural network classifier may not always match whatever makes it difficult to a human observer. Here, we notice a similar behavior, albeit in the context of language modeling.

Q3. Effect of STEPPED vs. BINNED

This experiment compares the effect of the two strategies of accessing the bins during training. Recall that in BINNED, the bins are disjoint and training samples are randomly selected from only one bin during a given epoch. On the other hand, in STEPPED, at each epoch, a new bin is added to the training set in addition to the previous bin, thereby, increasing the size of the training dataset progressively. For a fair comparison between the two approaches, we iterate over the VANILLA and BINNED models twice to reach the same number of steps as the STEPPED models.

Table 5 reports the results of this experiment allowing us to make the following key observations: (i) the BINNED (HARD) model yields the most competitive results, which is somewhat surprising given that one might consider such a strategy of training over non-overlapping bins to perhaps lead to catastrophic forgetting. To further analyze this phenomenon, when we compute the ratio of common word types between any two consecutive bins, we find an overlap of up to 40%, which could possibly explain why disjoint training works just as well as progressively incremental training; (ii) once again, we observe that adopting a document-level anti-curriculum learning, i.e., hard to easy sequence is beneficial especially after fine-tuning; and, (iii) the disconnect between the intrinsic and extrinsic evaluation results continues to persist.

Q4. What about Sustainability?

In Table 6 we summarize the results of the top performing model and the vanilla training runs in order to compare them in terms of the compute power consumed, i.e., the number of training steps, along with their validation loss and average GLUE score. We find that while, unsurprisingly, VANILLA model improves given more epochs, it still does not match the performance afforded by the BINNED model which is reached in almost half the number of steps[7], and this we find to be one of the most interesting findings of this study. This is particularly encouraging because it suggests that simple corpus ordering can not only improve the quality of the language model, but also its efficiency in terms of

[7] Even though the same base corpus is used for all the experiments, the slight difference in the number of steps is due to the way data is randomly sampled and chunked into input sequences during training.

Model	Loss ↓	GLUE ↑	# steps ↓
VANILLA (1 epoch)	**6.09**	62.2	28.2K
VANILLA (2 epochs)	**5.98**	63.4	56.4K
BINNED$_{\text{FRE}}$ (HARD) (1 epoch)	6.11	**63.9**	32.1K

Table 6: Compute resources in terms of training steps.

the training steps needed to reach a certain level of performance after fine-tuning.

We acknowledge that this work's conclusions are largely drawn based on experiments run on a reasonably sized transformer language model over a relatively small corpus[8] and a handful of epochs. We hope that future research can build upon these simple yet effective ideas.

5 Related Work

There is a growing awareness of the benefits of Green AI (Schwartz et al., 2020; Bender et al., 2021), with an emphasis on not just accuracy, but also efficiency, with the desirable goal, amongst many, of decreasing the environmental carbon footprint of pretrained language models, during training or inference. Tremendous progress is being made towards sustainable or 'greener' models with many efforts in improving model efficiency in order to reduce the amount of compute required (Hinton et al., 2015; Sanh et al., 2019; Jiao et al., 2020; Mao et al., 2020; Anderson and Gómez-Rodríguez, 2020; Schick and Schütze, 2021; Stock et al., 2021; Iandola et al., 2020).

Scaling laws (Kaplan et al., 2020) suggest that optimal compute-efficient training involves training very large models on a relatively modest amount of data and indeed, many studies explore the relationship between the volume of training data and consequent model performance (Banko and Brill, 2001; Sun et al., 2017; van Schijndel et al., 2019; Hu et al., 2020; Raffel et al., 2020; Brown et al., 2020; Zhang et al., 2021), generally concluding that rapid improvements in performance are observed as the amount of training data increases, at least until a certain point, after which the improvements slow down.

In this work, our focus is on studying the role of *corpus ordering* in language modeling, and although a handful of works explored this idea via length-based metrics (Bengio et al., 2009; Graves et al., 2017), they did not integrate document-level measures of complexity, nor did they study this in the context of transformer models. Furthermore, we perform an extensive evaluation reporting metrics pertaining to both pretraining and fine-tuning stages.

Prior work has typically measured the complexity of text in terms of the length of a sequence (Spitkovsky et al., 2010; Cirik et al., 2016; Kocmi and Bojar, 2017; Subramanian et al., 2017; Zhang et al., 2018; Platanios et al., 2019; Chang et al., 2021). Other metrics include features such as diversity, simplicity, and prototypicality (Tsvetkov et al., 2016), or the norm of a word embedding (Liu et al., 2020). The effects of these prior efforts have mostly been studied at the fine-tuning stage. At the pretraining stage, Babanejad et al. (2020) studied the effects of preprocessing training corpora in the context of affective tasks, while more recently, Press et al. (2021) showed that initially training a model on shorter subsequences before moving onto longer ones has benefits in terms of training time and model perplexity, although they did not study the impact on downstream tasks.

Although in regular settings curriculum learning has often led to mixed results, under limited training budgets, curriculum has shown to improve the performance (Cirik et al., 2016; Wu et al., 2020), which motivated us to explore the effects of corpus ordering in the task of transformer-based language modeling, and evaluating the metrics related to not just accuracy but also training efficiency as measured in the number of training steps.

6 Conclusions and Future Directions

In this paper, we investigate several corpus ordering and training schedules as a way of exploring the effectiveness of curriculum learning, specifically with regards to obtaining better models and potentially reducing training time and/or cost, for pretraining transformer language models (English, in this case). Between the two notions of complexity, one computed at sentence-level, and the other at document-level, we find them to perform relatively comparably. In the document-level curriculum, our findings suggest that going from hard to easy training samples may be an effective strategy. Furthermore, iterating over disjoint bins one at a time seems comparably effective to incrementally increasing the training data size. Finally, interestingly, our empirical results on eight down-

[8] 103.2 million tokens of WikiText-103 corpus as compared to some recent models leveraging upto 130 billion words (Linzen, 2020).

stream tasks from GLUE benchmark reveal that an ordered corpus yields competitive performance as compared to vanilla training in almost half the number of training steps, as measured over two epochs of training.

Many interesting avenues for future work remain such as devising more efficient corpus ordering algorithms or verifying whether these simple yet effective strategies generalize to different training corpora or model architectures, thus enabling development of sustainable language models.

Acknowledgments

We thank the anonymous reviewers for their insightful comments.

References

Mark Anderson and Carlos Gómez-Rodríguez. 2020. Distilling neural networks for greener and faster dependency parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 2–13, Online. Association for Computational Linguistics.

Nastaran Babanejad, Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. A comprehensive analysis of preprocessing for word representation learning in affective tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5799–5810.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 26–33.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

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.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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 *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 727–733, Online. Association for Computational Linguistics.

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*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

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.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Forrest Iandola, Albert Shaw, Ravi Krishna, and Kurt Keutzer. 2020. SqueezeBERT: What can computer vision teach NLP about efficient neural networks? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 124–135, Online. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

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.

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*, Seattle, Washington. 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.

Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Quanlu Zhang, Yaming Yang, Yunhai Tong, and Jing Bai. 2020. LadaBERT: Lightweight adaptation of BERT through hybrid model compression. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3225–3234, Barcelona, Spain (Online). International Committee on Computational Linguistics.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5493–5505, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commununications of the ACM*, 63(12):54–63.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum learning: A survey. *arXiv preprint arXiv:2101.10382*.

Valentin I Spitkovsky, Hiyan Alshawi, and Dan 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.

Pierre Stock, Angela Fan, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand

Joulin. 2021. Training with quantization noise for extreme model compression. In *International Conference on Learning Representations*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 241–251.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. 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 5831–5837, Hong Kong, China. Association for Computational Linguistics.

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.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2020. A comprehensive survey on curriculum learning. *arXiv preprint arXiv:2010.13166*.

Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pages 5238–5246. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2020. When do curricula work? *arXiv preprint arXiv:2012.03107*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

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*.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

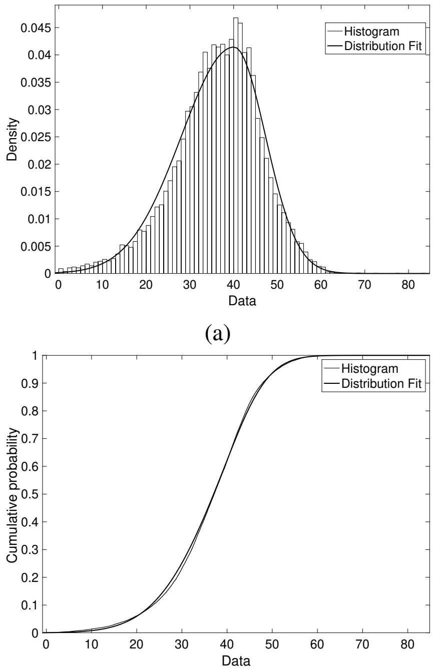
Appendix

A Sample Examples from Different Text Complexity Bins

Tables 7 and 8 present some sample text excerpts from *easy* and *hard* bins, as computed by document-level complexity and sentence-level complexity, respectively.

B Flesch Readability Ease

We computed the FRE value for each document in WikiText and create a histogram and fit a distribution, Figure 2. For WikiText, the *Epsilon-Skew-Normal* distribution provides the best fit with a mean FRE score of 36.23 and a standard deviation of 9.7.



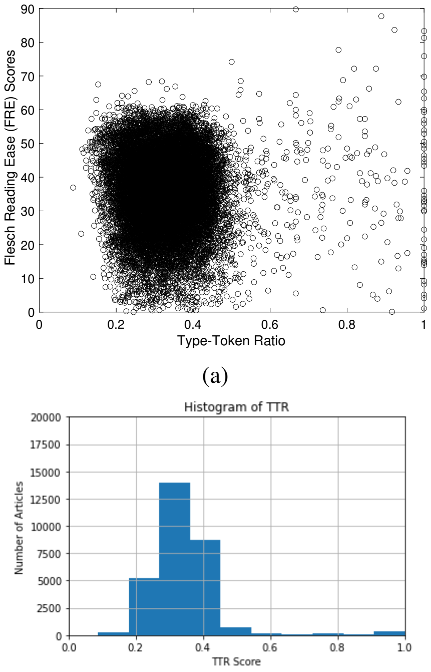
(a)

(b)

Figure 2: (a) Distribution fit and histogram, and (b) Cumulative distribution function, of the FRE scores as computed on the WikiText-103 corpus. We see that 70% of the documents have a FRE score in a narrow range between 35 and 55. We use this analysis to create the curriculum described in the text.

C Type Token Ratio

In our preliminary experiments, we also considered another document-level metric of complexity, namely, type token ration (TTR) which takes into account the unique number of word types in addition to the total number of word tokens, which may guide in creating more meaningful bins. Figure 3(a) indicates little correlation between FRE and TTR, which further motivated us to try this line of investigation, whereas Figure 3(b) shows the histogram of TTR scores which was used to identify bin boundaries. During the experiments, however, TTR's performance did not match that of FRE.



(a)

(b)

Figure 3: (a) Flesch Reading Ease vs. TTR score, and (b) Histogram of the TTR scores as computed on the WikiText-103 corpus.

Complexity	Text
EASY	A last-minute addition to the film, the simple love song was quickly written by Ashman and Menken to replace the more elaborate and ambitious "Human Again" after the latter was cut from Beauty and the Beast. O'Hara based her own vocal performance on that of American singer and actress Barbra Streisand, who Howard advised the actress to impersonate, while O'Hara herself convinced the songwriters to have Benson record the song. Critical reception towards "Something There" has been positive, with film and music critics alike praising Ashman's abilities as both a songwriter and a storyteller.
HARD	TM and Cult Mania is a non-fiction book that examines assertions made by the Transcendental Meditation movement (TM). The book is authored by Michael Persinger, Normand Carrey and Lynn Suess and published in 1980 by Christopher Publishing House. Persinger is a neurophysiologist and has worked out of Laurentian University. He trained as a psychologist and focused on the impacts of religious experience. Carrey is a medical doctor who specialized in psychiatry. He focused his studies into child psychiatry with research at Dalhousie University, and has taught physicians in a psychiatry residency program in the field of family therapy. Suess assisted Persinger in researching effects of geological phenomena on unidentified flying object sightings in Washington; the two conducted similar research in Toronto and Ottawa.

Table 7: Easy and hard text samples as measured by FRE scores at document level.

Complexity	Text
EASY	(i) Most Egyptian deities represent natural or social phenomena. (ii) But some deities represented disruption to maat. (iii) Not all aspects of existence were seen as deities. (iv) Divine behavior was believed to govern all of nature. (v) In myth, the gods behave much like humans. (vi) Some have unique character traits. (vii) Gods were linked with specific regions of the universe. (viii) Temples were their main means of contact with humanity. (ix) The gods were believed to have many names. (x) This divine assemblage had a vague and changeable hierarchy.
HARD	(i) Employing the same fusion of tactical and real-time gameplay as its predecessors, the story runs parallel to the first game and follows the "Nameless", a penal military unit serving the nation of Gallia during the Second Europan War who perform secret black operations and are pitted against the Imperial unit "Raven". (ii) Characters also have Special Abilities that grant them temporary boosts on the battlefield: Kurt can activate "Direct Command" and move around the battlefield without depleting his Action Point gauge, the character can shift into her "Valkyria Form" and become invincible, while Imca can target multiple enemy units with her heavy weapon. (iii) The three main characters are No.7 Kurt Irving, an army officer falsely accused of treason who wishes to redeem himself; Ace No.1 Imca, a female Darcsen heavy weapons specialist who seeks revenge against the Valkyria who destroyed her home; and No.13 Riela, a seemingly jinxed young woman who is unknowingly a descendant of the Valkyria. (iv) Speaking in an interview, it was stated that the development team considered Valkyria Chronicles III to be the series 'first true sequel: while Valkyria Chronicles II had required a large amount of trial and error during development due to the platform move, the third game gave them a chance to improve upon the best parts of Valkyria Chronicles II due to being on the same platform.

Table 8: Easy and hard text samples as measured by sentence length. We hypothesize that sentence shuffled corpus still yields competitive results as compared to document-level ordering is because in general, the larger context surrounding the sentences continues to be maintained as shorter sentences from the same documents remain together, and longer sentences from the same document cluster together. It is worth noting that the sentences in this table are delineated just for presentation purposes. In actual implementation, the sentences are all combined as one (very) long document for efficient processing.