Do Data-based Curricula Work?

Maxim K. Surkov **Vladislav D. Mosin** **Ivan P. Yamshchikov**
LEYA Lab, Yandex, Higher School of Economics

Abstract

Current state-of-the-art NLP systems use large neural networks that require extensive computational resources for training. Inspired by human knowledge acquisition, researchers have proposed curriculum learning - sequencing tasks (task-based curricula) or ordering and sampling the datasets (data-based curricula) that facilitate training. This work investigates the benefits of data-based curriculum learning for large language models such as BERT and T5. We experiment with various curricula based on complexity measures and different sampling strategies. Extensive experiments on several NLP tasks show that curricula based on various complexity measures rarely have any benefits, while random sampling performs either as well or better than curricula.

1 Introduction

In the last years state-of-art results in natural language processing (NLP) are often obtained with Transformer-like architectures based on the self-attention mechanism (Vaswani et al., 2017) such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2020), which could have billions of parameters. Due to many parameters, these architectures require lots of time and hardware resources to be trained.

Curriculum learning (CL) is one of the popular methods to reduce training time and increase the resulting quality of the model. Inspired by the importance of adequately ordering information when teaching humans (Avrahami et al., 1997), curriculum learning increases the difficulty of training samples shown to the model over time (Elman, 1993). Previous studies have demonstrated that curriculum learning significantly impacts training time and quality in different machine learning domains, such as computer vision (Soviany, 2020) and reinforcement learning (Narvekar et al., 2020). In NLP, some results hint that CL might be beneficial (Platanios et al., 2019; Xu et al., 2020; Kocmi and Bojar, 2017); however, these results are not as optimistic as in reinforcement learning setup.

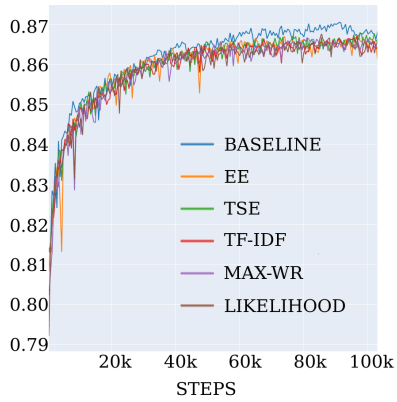
We suggest dividing recent research in curriculum learning into two main categories: *task-driven* curriculum and *data-driven* curriculum. The idea of the task-driven curriculum was inspired by human behavior. First, the model learns how to solve a simple task, and then the difficulty is gradually increased. This type of curriculum proposed by Bengio et al. (2009) is considered to be classical, and a majority of curriculum-related results are obtained in this framework. Alternatively to the task-driven curriculum, some curricula try to use some form of filtering or sorting of training data that could facilitate learning a model on a given task. We suggest calling these curricula *data-driven* and distinguishing them from the classical task-based approach.

This paper attempts to understand when data-driven curriculum learning works for transformer-based language models. Generally, data-driven curriculum learning is organized in two steps: first, estimating the complexity for the elements that comprise the dataset; second, designing a sampling strategy, thus forming a curriculum. In the first part of the paper, we list potentially useful natural language processing complexity measures. The second part discusses possible sampling strategies that might apply to corresponding complexity measures. We run extensive experiments with different metrics and sampling strategies on three classes of NLP tasks: unsupervised learning with masked language modeling, text classification, and machine translation. Our experiments show that data-driven curriculum learning does not give quality increase or time reduction on all metric-sampling strategy setups and often makes results even worse.

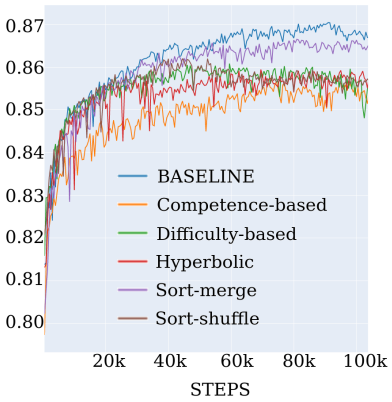
2 Metrics

The first important part of the curriculum learning pipeline is measuring the complexity of samples

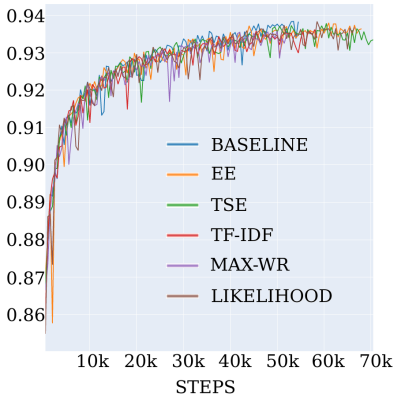
Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 119 - 128



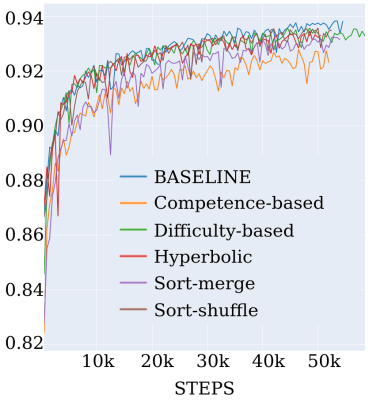
(a) Sentiment140 with sort-merge sampler for all complexity measures.



(b) Sentiment140 with max word rank complexity measure for all samplers.



(c) Hyperpartisan News with sort-shuffle samples for all complexity measures



(d) Hyperpartisan News with max word rank complexity measure for all samplers.

Figure 1: Pre-trained BERT fine-tuned on Sentiment140 and Hyperpartisan News Detection datasets. Accuracy of the classifier as a function of the number of training steps.

for a given dataset. Texts could have a complex structure, and one can measure their complexity in different ways. A variety of heuristically motivated methods is accompanied by several metrics based on specific aspects of information theory. For a review of heuristic text complexity measures such as length of TF-IDF (Aizawa, 2003) we address the reader to Appendix A. In this paper, we also explore the metrics initially proposed by Ay et al. (2006) to measure the complexity of finite systems and try to see if one could apply these metrics to NLP tasks.

Ay et al. (2006) observes that for finite systems, a set of parts impacts the complexity of the system as well as inter-dependencies of the parts. In the context of NLP, this means that text is more than just a bag of words. The authors propose four different metrics to estimate the complexity of a system. However, one of these metrics maximizes on single-letter texts, such as "Aaaaaaaaa," while the second was created to measure cyclic sequences and does not apply to texts. Thus we experiment with two other metrics, namely, Tononi, Sporns, and Edelman (TSE) (Tononi et al., 1994) and excess entropy (EE), and adapt them to the complexity of texts. For the calculation of TSE and EE for NLP we address the reader to Appendix B.

3 Samplers

The second important part of curriculum learning is the sampling strategy (or sampler) - the algorithm deciding which samples should be shown to the model at which moment. Let us observe existing curricula and suggest some new ones.

Competence-based. CB
A competence-based curriculum, offered by Platanios et al. (2019), uniformly samples data from increasing dataset's prefix. Competence is a function $c(t)$, which defines the size of the dataset prefix.

$$c(t) = \min\left(1, \sqrt{t\frac{1-c_0^2}{T} + c_0^2}\right)$$

Where T - total number of steps, t - current step, c_0 - hyperparameter set to 0.01.

Hyperbolic. HYP

The main idea of this sampler is to increase average batch complexity through time. All samples are split by complexity into N sequential buckets with equal size. Training time is divided into N epochs and the probability of sampling the element from the j-th bucket on the i-th epoch is proportional to the distance between j and i.

$$Pr_i(j) = \frac{c}{|j-i|^{0.5}}$$

Where $Pr_i(j)$ - probability to sample from j-th bucket on the i-th epoch, c - constant to guarantee that sum of all probabilities equals to 1.

Difficulty-based. DB

This sampler is a reversed version of the competence-based one. A difficulty-based sampler takes elements from a linearly decreasing suffix instead of sampling from a gradually increasing prefix.

Sort-shuffle. SS

All previously described samplers do not guarantee that the model would see each element in the training data. Sort-shuffle samples each element exactly once, randomly splitting the data into batches and sorting by average complexity.

Sort-merge. SM

Many complexity estimates correlate with the length of the text. The main idea of a sort-merge sampler is to remove this correlation and train the model on stable length distribution. This algorithm consists of four main steps: sort dataset by length; sequentially split into buckets; sort each bucket by a complexity metric; form i-th batch from i-th elements from each bucket. Like a sequential one, the sort-merge sampler shows each element to the model exactly once.

Equipped with the list of metrics and curriculum samplers, we can discuss our experimental results.

4 Experiments

We perform our experiments on three NLP tasks: text classification, machine translation (NMT), and masked language modeling (MLM). Here we discuss the first task of classification in detail. The extensive results of the experiments are available in Appendix C. All the experiments are performed with the HuggingFace library (Wolf et al., 2020), which provides the models with their setups, such as hyperparameters and tokenizers. We did not change default parameters in our experiment unless specifically stated otherwise. Thus, the dataset and the model specify every experiment. We use the base version of the BERT model (Devlin et al., 2019) for MLM and classification, and the small version of the T5 model (Raffel et al., 2020) for machine translation. Experiments were performed on BooksCorpus[1] dataset for MLM, Sentiment140[2] and Hyperpartisan News Detection[3] for classification, and WMT16-en-de[4] for machine translation. To estimate the curriculum's convergence speed, we calculate the average number of steps to reach a threshold that is 10% lower than the resulting saturation quality metric for every problem.

4.1 Text Classification

Figure 1 summarizes the experiments with BERT for text classification. Neither different samplers nor complexity measures improve a BERT-based classifier's resulting accuracy.

4.2 Masked Language Modelling

Figure 2 shows the results of MLM pretraining of BERT on BooksCorpus. Irrespective of sampling, the complexity measures have similar ranking in terms of their performance on MLM: length, likelihood, TSE, EE, TF-IDF, maximum word rank. Since sorted sampler takes length into account by design, it is not included in the corresponding plots. Data-based curricula show inferior results in comparison with the baseline.

4.3 Neural Machine Translation

Table 1 shows the experiments with T5 model (Raffel et al., 2020) for machine translation and various curricula. We use the BLEU metric to estimate the quality of the resulting models. We calculate the average BLEU score over ten validations at saturation. Once again, curriculum learning does not give any notable benefits.

5 Discussion

We try to interpret obtained results cautiously. Though Platanios et al. (2019) report that

[1] https://huggingface.co/datasets/bookcorpus

[2] https://www.kaggle.com/kazanova/sentiment140

[3] https://huggingface.co/datasets/hyperpartisan_news_detection

[4] https://huggingface.co/datasets/wmt16

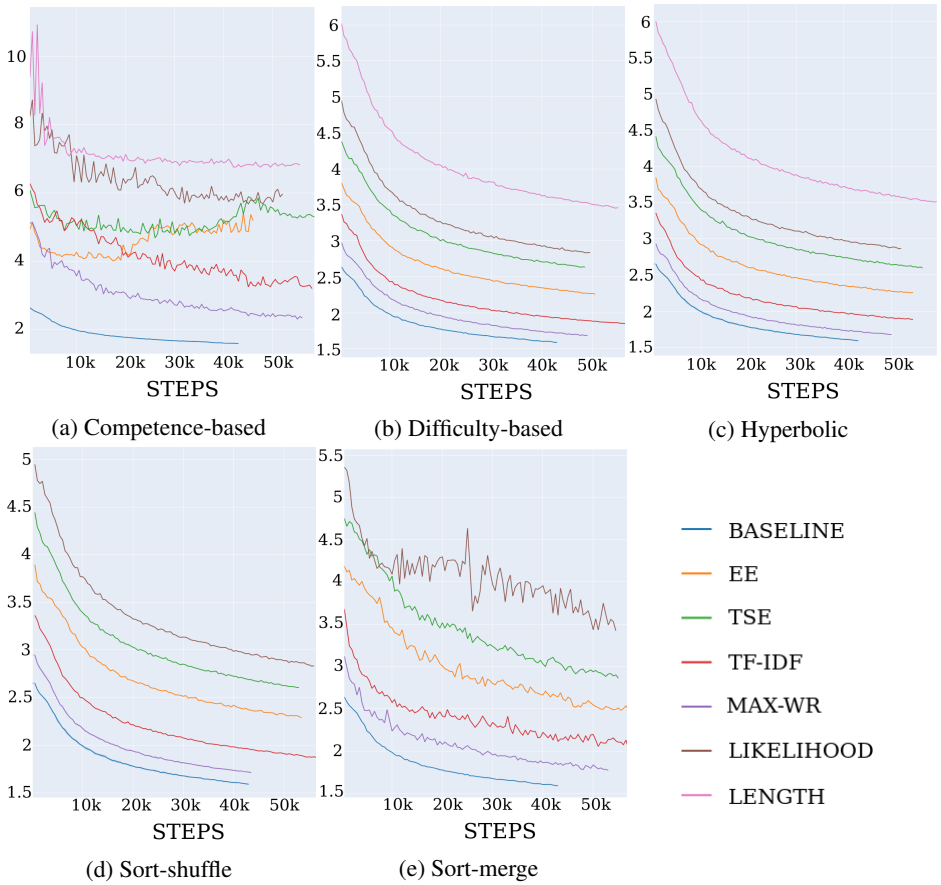


Figure 2: Loss function dependency on the number of training steps on MLM for BooksCorpus dataset during the first 40k steps of training. Every plot depicts results for six different complexity estimates combined with a specific sampler.

Table 1: The average BLEU score from 50k to 100k steps on WMT16 dataset. Results better than the baseline are highlighted. '-' denotes the cases when complexity measure and sampler are not compatible.

Metrics	Samplers				
	CB	DB	Hyp	SS	SM
baseline			18.3		
length	10.1	17.4	16.3	-	-
TSE	10.3	**18.4**	16.8	13.8	14.8
EE	10.2	18.2	16.9	13.3	15.0

competence-based sampling is beneficial for recurrent neural networks, we could not reproduce this result in transformer-based architectures. We also run experiments to check whether data-based curricula could work on non-transformer architectures. The results do not look encouraging; see Appendix C.2.

Curriculum learning depends on subtle factors, for example, a correct choice of hyperparameters. It is hard to check all possible values of hyperparameters, yet to the best of our capabilities, we address this issue in Appendix C.3. The results do not seem to depend on the learning rate, and once again, curriculum learning shows no benefits.

At this point, we can only conclusively say two things: (1) a deeper investigation of the underlying information theoretic principles that stand behind curriculum learning is badly needed; (2) until we better understand these principles, data-based curriculum learning is a gamble with very low odds to gain either speed or resulting performance.

6 Conclusion

In this work, we ran extensive experiments with curriculum learning for transformer-based architectures on three NLP tasks: masked language modeling, text classification, and machine translation. We demonstrate that curricula do not help in the standard training setting and sometimes even worsen results.

7 Acknowledgments

The publication was supported by the grant for research centers in the field of AI provided by the Analytical Center for the Government of the Russian Federation (ACRF) in accordance with the agreement on the provision of subsidies (identifier of the agreement 000000D730321P5Q0002) and the agreement with HSE University No. 70-2021-00139. This research was supported in part through computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021)

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Judith Avrahami, Yaakov Kareev, Yonatan Bogot, Ruth Caspi, Salomka Dunaevsky, and Sharon Lerner. 1997. Teaching by examples: Implications for the process of category acquisition. *The Quarterly Journal of Experimental Psychology Section A*, 50(3):586–606.

Nihat Ay, Eckehard Olbrich, Nils Bertschinger, and Jürgen Jost. 2006. A unifying framework for complexity measures of finite systems. In *Proceedings of ECCS*, volume 6. Citeseer.

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 Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

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

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.

PS Kostenetskiy, RA Chulkevich, and VI Kozyrev. 2021. Hpc resources of the higher school of economics. In *Journal of Physics: Conference Series*, volume 1740, page 012050. IOP Publishing.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50.

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.

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*, 21:1–67.

Petru Soviany. 2020. Curriculum learning with diversity for supervised computer vision tasks. In *MRC@ECAI*.

Giulio Tononi, Olaf Sporns, and Gerald M Edelman. 1994. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences*, 91(11):5033–5037.

Frans van der Sluis and Egon L van den Broek. 2010. Using complexity measures in information retrieval. In *Proceedings of the third symposium on information interaction in context*, pages 383–388.

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

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

Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

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*.

A Heuristic Approaches to Text Complexity

The first idea is to determine the complexity of the text as its length. Despite its simplicity, this method is used in different works (Platanios et al., 2019; Kocmi and Bojar, 2017). The next family of approaches boils down to phonological, morphological, lexical, or syntactic metrics derived with some form of expert linguistic knowledge. However, van der Sluis and van den Broek (2010) used Wikipedia and Simple Wikipedia corpora to demonstrate that language-based metrics do not correlate with the common sense text complexity. The third class of methods treats text as a bag of words and builds metrics based on the frequency analysis. For example, every word gets a rank equal to its position in the dictionary sorted by the number of word appearances in a corpus. In this case, complexity may be measured as a maximum rank among the words in a bag (Kocmi and Bojar, 2017). This metric is called max frequency rank. Another possible metric is called likelihood. The metric calculates the probability of the text under the assumption that all tokens are independent, just by multiplying probabilities of all tokens in the text (Platanios et al., 2019). Another metric from this group is TF-IDF (Aizawa, 2003), which is widely used in search systems. Finally, the last array of methods is based on using different neural network losses as a complexity measure of a sample.

B Using Information Theory for Text Complexity

Let $X_V = (X_{v1}, X_{v2}, \ldots)$ be a sequence of random variables from set $V = (v1, v2, \ldots)$, and A is a subset of V, then X_A is a subsequence of X_V with elements from A. Let's determine $H(X_A)$ as entropy of sequence X_A. However, texts consist of words or tokens, not random variables. We propose the following procedure of transforming texts into random variable sequences. For each token in position i we compute the percentage of texts with this token on the same position and replace the original token with binary distribution with a probability of one equal to the calculated percentage. After transforming text into a sequence of random variables, we can compute its entropy.

$$\begin{aligned} H(X_V) &= H(X_{v1}) + H(X_{v2}|X_{v1}) \\ &+ H(X_{v3}|X_{v2}, X_{v1}) + \ldots \end{aligned}$$

If one wants to apply this formula, one must compute entropy for many different conditional distributions while these distributions depend on the order of tokens in a text. First, direct application of the formula would overfit a specific text since all texts are different in a corpus. Second, such computation could not be carried out in a reasonable time. The limit context for conditional distributions to the nearest neighbors one obtains the following formula

$$H(X_V) = H(X_{v1}) + \sum_{i=2}^{\#V} H(X_{v_i}|X_{v_{i-1}})$$

Using this approximation for entropy one can compute excess entropy (EE) and the complexity measure Tononi, Sporns and Edelman (TSE), (Tononi et al., 1994) as they are formulated by Ay et al. (2006)

$$EE(X_V) = \left[\sum_{v \in V} H(X_{V \setminus v})\right] - (n-1)H(X_V), \quad (1)$$

$$TSE(X_V) = \sum_{k=1}^{n-1} \frac{k}{n} C^{(k)}(X_V), \quad (2)$$

where n is a size of set V and

$$C^{(k)}(X_V) = \frac{n}{k\binom{n}{k}} \sum_{A \subseteq V, |A|=k} H(X_A) - H(X_V).$$

C Additional Experiments

C.1 Convergence Speed

Curriculum learning is often apprised for the speed-up of the model's convergence. The intuition here is to provide a curriculum that would help to achieve the same result faster, yet without a significant loss in quality. We carried out several experiments to see if data-based curricula could speed up the learning in transformer-based language models.

C.1.1 Classification

Tables 2 3 show average number of training steps needed to reach 90% of the resulting accuracy for the corresponding classification task. On Sentiment140 TF-IDF, TSE, and maximum word rank speed the convergence up to 3% with some samplers. However, other metrics or sampling strategies slow down the model's convergence speed, while on a bigger HND dataset, other curricula show results better than the baseline. One could conclusively say that length is the worse metric to organize curriculum in all experiment configurations. The one more important conclusion is that the model can not always estimate the complexity of the sample concerning its' internal state (MLM-loss does not speed up the training speed and drawdown the final model quality on the Sentiment140 dataset). This happens when the model is expressive enough, and all samples have equal complexity in model-based metrics.

C.1.2 Pretraining MLM

Figure 2 shows a significant slowdown in model convergence speed can be seen for all curricula compared to the baseline learning regime. One can also divide all metrics into two distinct groups. The first one consists of maximum word rank and TF-IDF. The second group includes EE, TSE, likelihood, and length. The metrics in the first group allow the model to converge to a lower loss value. However, the second group's metrics hinder the convergence and seem to have higher saturation loss. Hence, it isn't easy to find a universal threshold to reasonably compare all metrics and samplers. One should also note that only maximum word rank does not degrade the model quality compared to the baseline, while other curricula cause severe deterioration. Finally, the last main observation is that curriculum learning, unfortunately, does not allow us to run MLM faster. Moreover, the number of training steps needed to reach a given threshold could be several times higher in comparison with the baseline approach. Table 4 illustrates this fact.

C.2 Data-based Curricula for Other Architectures

It seems that data-based curriculum learning cannot increase quality or reduce training time for transformer-based models. Though Platanios et al. (2019) report that competence-based sampling is beneficial for recurrent neural networks, we could not reproduce this result in transformer-based architectures. While some curricula might be useful for smaller architectures on some tasks, they have no significant benefits for larger architectures. Let us double-check that with the recurrent neural network architecture to see if the negative result obtained above is associated with certain properties of attention-based architectures or could be reproduced with various artificial neural networks. We run our experiments on Sentiment 140 with 90% train and 10% test split. The curricula include Hyperbole, Difficulty-Based and Competence-Based samplers, and TSE and length difficulty metrics. Figure 3 shows that data-driven curricula do not have a significant influence on the results.

Comparing Figure 3 with Tables 3 – 2 one could see that data-based curricula are hardly beneficial even for smaller architectures. Rather, under certain conditions, one could get some improvement of convergence, yet on a different task, the same choice of complexity measure and sampling strategy would be on par with the baseline.

C.3 Data-based curricula and Hyperparameters

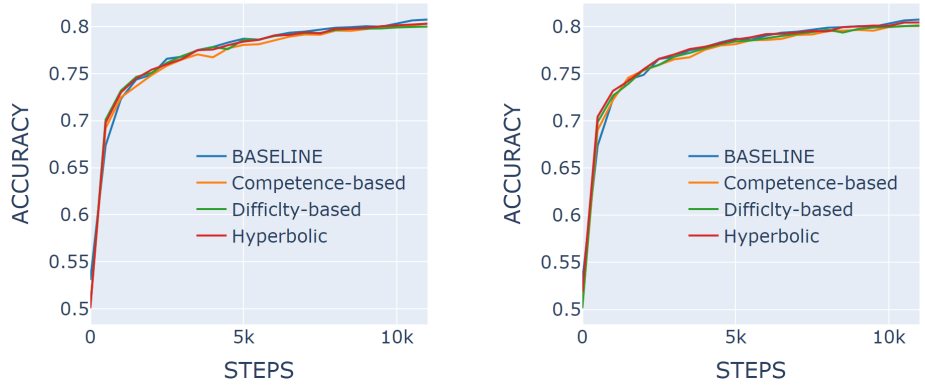
Extensive experiments on different NLP tasks show that data-based curriculum learning does not help to increase quality with default hyperparameters. Hyperparameters' importance for the curriculum is an open question. Some papers state that hyperparameters, especially learning rate, are essential for curriculum (Zhang et al., 2018). On the other hand, some papers propose methods that are not highly sensitive to hyperparameters (Platanios et al., 2019). It seems that hyperparameters choice is discussed mainly in the works addressing NMT, so we run additional experiments with our curricula and three different learning rates (10^{-3}, 10^{-4}, 10^{-5}) on NMT as well. Results demonstrate that models' behavior does not depend on the learning rate much, and for every learning rate, curricula do not give a significant quality increase. Results for excess entropy are presented in Figure 6.

Table 2: The average number of steps needed to reach given threshold for all configurations metric-sampler on text classification task on Hyperpartisan News Detections dataset. Maximal deviation for 3 runs is less than $3k$ steps. Results better than the baseline are highlighted. ∞ means that model did not reach the threshold, '-' denotes the cases when complexity measure and sampler are not compatible.

Metrics	Threshold	Accuracy	Samplers				
			CB	DB	Hyp	SS	SM
baseline	92.9%	93.8%			22k		
length	92.9%	93.7%	55k	23k	22.5k	-	-
TF-IDF	92.9%	93.5%	∞	**19.5k**	24k	23.5k	33k
TSE	92.9%	93.8%	56.5k	21k	23k	22k	31k
EE	92.9%	93.8%	71.5k	25.5k	22.5k	**19.5k**	32.5k
max wr	92.9%	93.6%	∞	22k	**20.5k**	22.5k	39k
likelihood	92.9%	93.8%	∞	**20k**	24k	**20k**	30k
MLM-loss	92.9%	**93.9%**	23.5k	**18k**	23k	24k	**20k**

Table 3: The average number of steps needed to reach given threshold for all configurations metric-sampler on text classification task on sentiment140 dataset. Maximal deviation for 3 runs is less than $3k$ steps. Results better than the baseline are highlighted. ∞ means that model did not reach the threshold, '-' denotes the cases when complexity measure and sampler are not compatible.

Metrics	Threshold	Accuracy	Samplers				
			CB	DB	Hyp	SS	SM
baseline	85.5%	**87%**			17.5k		
length	85.5%	86.2%	112.5k	20k	19k	-	-
TF-IDF	85.5%	86.7%	115.5k	21.5k	19.5k	**16.5k**	22k
TSE	85.5%	86.8%	95.5k	**16.5k**	20.5k	21.5k	18k
EE	85.5%	86.7%	59k	19.3k	23k	20k	19k
max wr	85.5%	86.7%	70k	18.5k	19.5k	**17k**	19k
likelihood	85.5%	86.7%	112k	17.5k	21.5k	17.5k	21.5k
MLM-loss	85.5%	86.1%	59.5k	21k	23.5k	19.5k	20k



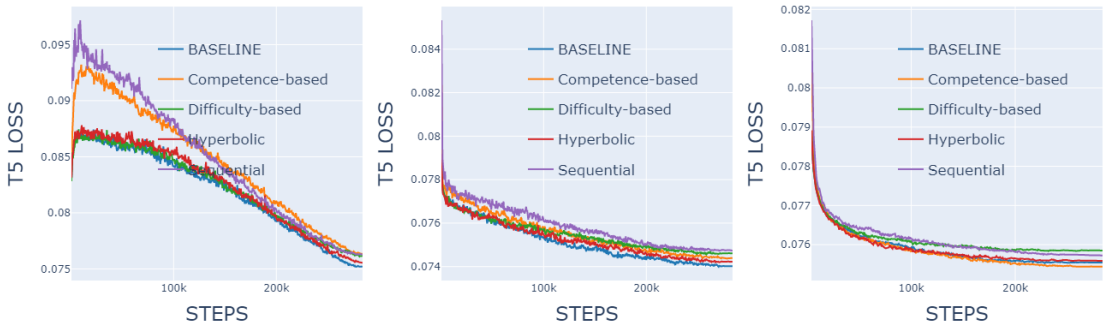
(a) Sentiment140 with length as complexity metric and three samplers.

(b) Sentiment140 with TSE as complexity metric and three samplers.

Figure 3: Test results with LSTM on Sentiment140 dataset. Accuracy of the classifier as a function of the number of training steps.

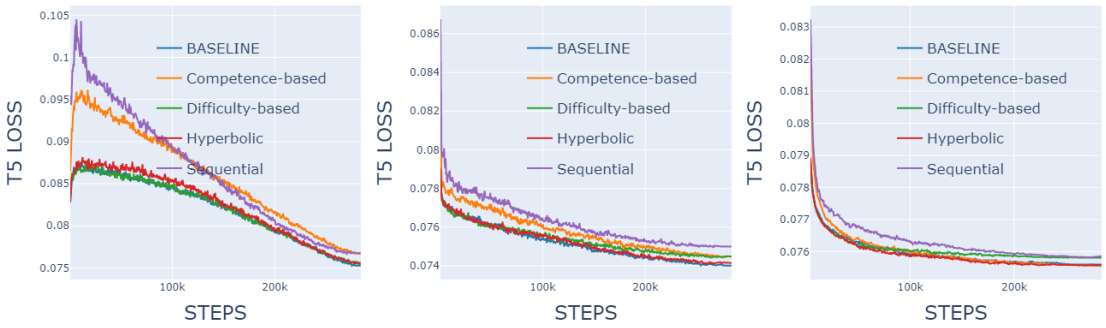
Table 4: The average number of steps needed to reach given threshold for all configurations metric-sampler on pretraining on BooksCorpus dataset. Maximal deviation for 3 runs is less than $3k$ steps. All complexity measures based curricula reach saturation at higher losses than the baseline thus we used an arbitrary threshold of 3.5 for them. Results better than the baseline are highlighted. ∞ means that model did not reach the threshold, '-' denotes the cases when complexity measure and sampler are not compatible.

Metrics	Threshold Loss	Saturation Loss	Samplers				
			CB	DB	Hyp	SS	SM
baseline	2.00	**1.58**			9.5k		
max wr	2.00	**1.58**	∞	17.5k	16.5k	16.5k	27k
TF-IDF	2.00	1.84	∞	34k	35k	37.5k	∞
EE	3.50	2.25	∞	4k	**3.5k**	4.5k	9.5k
TSE	3.50	2.60	∞	9k	9k	8.5k	18k
likelihood	3.50	2.83	∞	13.5k	13.5k	15.5k	50k
length	3.50	3.45	∞	50.5k	∞	-	-



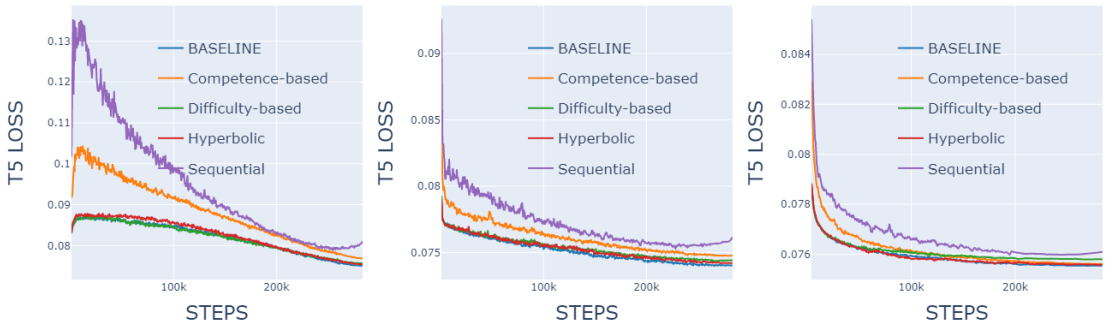
(a) learning rate 10^{-3} (b) learning rate 10^{-4} (c) learning rate 10^{-5}

Figure 4: Test results for NMT on WMT16 with different learning rates with excess entropy as a complexity measure



(a) learning rate 10^{-3} (b) learning rate 10^{-4} (c) learning rate 10^{-5}

Figure 5: Test results for NMT on WMT16 with different learning rates with TSE as a complexity measure



(a) learning rate 10^{-3} (b) learning rate 10^{-4} (c) learning rate 10^{-3}

Figure 6: Test results for NMT on WMT16 with different learning rates with length complexity measure