Byte-ranked Curriculum Learning for BabyLM Strict-small Shared Task 2023

Justin DeBenedetto
Villanova University

Abstract

The size of neural language models has increased rapidly over the past several years. This increase in model size has been accompanied by using larger and larger amounts of language data to train them. As these models and training data sizes have grown, the computational resources required to train them has surpassed what is available to many researchers. This work is part of a shared task called the BabyLM Challenge which requires language models to be trained using a restricted amount of training data a small fraction of the size of what large models use. In addition, no pretrained tools can be used. This work presents a curriculum learning approach to this data restricted setting by applying a bytes per line ordering to provided datasets. Throughout training, the average bytes per line is gradually increased by including more datasets as training data. Overall, there is an increase in performance on downstream tasks when using this curriculum learning approach, which provides a basis for potential further exploration of byte-based curriculum learning approaches.

1 Introduction

Large language models (LLMs) have received much attention from researchers and the general public in recent years (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; Hoffmann et al., 2022). One distinguishing aspect of these recent models is an explosion in the size of the models and a corresponding massive increase in training data to train these large models. In particular, the Chinchilla (Hoffmann et al., 2022) work suggests that model size and training tokens should be scaled at the same rate. To demonstrate the importance of the amount of training data used to train a model, Chinchilla was trained with 1.4 trillion training tokens, nearly five times the size of the training data for other LLMs at the time. The result was an improvement on a number of downstream tasks.

While large models perform very well on a large variety of tasks, they also come with many drawbacks. These models require large amounts of computing resources beyond what is available to many researchers. Additionally, the amount of data used to train these models is not currently available in the majority of the world's languages. In an effort to investigate language modeling abilities and training strategies in data-limited situations, the BabyLM challenge restricts the amount of data available to models (Warstadt et al., 2023).

One approach to improve training speed and improve downstream performance is by providing training data in a specific order. In particular, gradually increasing the difficulty of the training samples provided to the model is known as curriculum learning (Elman, 1993). Human children learning language follow a similar exposure to language. Speech directed at babies is far simpler than speech directed at adults and written language data follows the same trend. The motivation behind curriculum learning is to treat a neural network in a similar manner and allow it to learn from easier training samples before being presented with more difficult training samples.

The approach taken in this current work is to apply curriculum learning in a data restricted setting, without incorporating outside knowledge or data, to see its impact on training. The preprocessing steps are kept the same across models presented to reduce their effect on the ability to compare across training runs. A byte-level byte-pair-encoding tokenization is used across all models presented. Inspired by the byte-level approach to encoding, bytes per line is used as the measure of "difficulty" for a given portion of the dataset. The data used to train the model came from several different datasets. The bytes per line "difficulty" is used to determine the order in which training datasets are provided to the models as part of a curriculum learning approach. While no additional or outside information is required to apply this approach to data, the result for this challenge was that transcribed speech was used as training data

Proceedings of the 27th Conference on Computational Natural Language Learning: Volume 2: The BabyLM Challenge, pages 198–206

before any of the written text data. This provides another parallel to human language acquisition as speech comes before literacy in children.

Given that the limitations motivating this work and this challenge include data limitations as well as computational resources, we train each model for a set number of epochs. Models trained using the curriculum learning approach outperformed a traditional training approach baseline across several benchmark downstream tasks. When computational resources are less limited, the models also continue to improve when the model size is increased and when trained longer.

2 Related Work

There is much existing work on language models, including methods to work with them in computational or data constrained settings. One approach that has been used is model distillation. A well-known example of this is DistilBERT (Sanh et al., 2019). DistilBERT, and other distillation trained models, require a larger pretrained model to act as a teacher when a smaller model is trained. While the end result is a smaller model which can perform quite well. This approach can be applied to systems which require a small final model, but does not work for data or computationally constrained settings for training such as ours.

A similar approach is to use a large language model and simply finetune on the data-restricted task. Since this requires a pretrained large language model, this approach also does not work for constrained training settings with no such pretrained model available. While finetuning is used as part of the evaluation process for this challenge, this approach violates the restrictions of this challenge. As such, this solution to data-restricted settings is not used here. When there is a domain mismatch between the data used to pretrain an existing language model and the training data for a desired domain, some work suggests that training a new language model may be beneficial. For example, Gu et al. (2021) find that training a new language model specifically on in-domain biomedical data produced a better result for in-domain downstream tasks. This is more similar to the setting of this work as a language model is trained from scratch.

Another area of research within Natural Language Processing that is similar to this strict-small track is work with low-resource languages. While many of the largest language models are built for English with large quantities of data, there have been efforts to improve language modeling in lower resource language as well. Some of these, such as multilingual BERT (Devlin et al., 2018), are themselves large language models which combine many languages into one model. These models still require a large amount of resources (data and computational) and are larger than what is presented in the challenge.

Since curriculum learning relies upon increasing the difficulty of training samples as training continues, determining what makes a training sample more difficult than another is centrally important. For language input, some proposed measures of difficulty include presence of rare words (Bengio et al., 2009), block size (Nagatsuka et al., 2021), and length (Nagatsuka et al., 2023). When viewed in relation to these approaches, this work represents an exploration of a new, related measure of difficulty of training samples.

The learning schedule used in this work which determines at what rate new samples are added to the training set shares a similar motivation to work by Amiri et al. (2017). Their work applies findings from psychology that human learners learn effectively when the same information is reviewed with increasing lengths of time between reviews. These findings suggest that human learners ability to learn information is impacted not only by repetition of material, but also by the interval of time between those repetitions. The work by Amiri et al. (2017) uses this as a basis for a curriculum learning schedule. That work created a scheduler which spends more time on difficulty training instances and less time on easy instances. This work, by contrast, by gradually increasing the size of the training set, also gradually increases the time between repetitions of the easiest training samples while saving the more difficulty samples for later in training.

As this work was part of a shared task BabyLM challenge, there will be other related works published at the same time as this work. While those works cannot be discussed here, they will also provide good comparisons of other possible approaches.

3 Data

The dataset provided for this challenge came from ten sources. These sources were chosen to represent the type of language that a human child may be exposed to when learning English and includes both written text and transcribed speech. For the strict-small track, the total training data available was just under 10 million words.

Given the variety of sources, the text format was not consistent across the provided data and required some preprocessing.

3.1 Preprocessing

Due to the strict nature of the challenge, no preprocessing steps which were pretrained on outside data were allowed. This restriction ruled out the use of many off-the-shelf preprocessing tools. In many

Dataset	Domain	Words	Size (MB)	Lines	Bytes/line
CHILDES	Child-directed speech	0.44M	1.9	80K	24
OpenSubtitles	Movie subtitles	3.09M	16.0	527K	30
Switchboard Dialog Act Corpus	Dialogue	0.12M	0.6	16K	37
British National Corpus, dialogue portion	Dialogue	0.86M	4.3	89K	48
QCRI Educational Domain Corpus (QED)	Educational video subtitles	1.04M	5.6	100K	56
Simple Wikipedia	Wikipedia (Simple English)	1.52M	8.7	120K	72
Children's Book Test	Children's books	0.57M	2.6	26K	100
Standardized Project Gutenberg Corpus	Written English	0.99M	5.5	54K	102
Children's Stories Text Corpus	Children's books	0.34M	1.8	16K	112
Wikipedia	Wikipedia (English)	0.99M	5.8	50K	117
Total		9.96M	52.8	1078K	49

Table 1: Dataset provided for the strict-small track of the BabyLM challenge. Dataset names, domain descriptions, and word counts provided in Warstadt et al. (2023). Bytes, line counts, and bytes per line all measured after preprocessing was completed. See section 3.1 for details.

low-resource settings there may be no or limited existing pretrained tools to use for preprocessing. While such tools are useful when available, in this challenge those tools are off-limits.

We used a rule-based sentence splitter. Sentences are automatically split by punctuation unless they are preceded by one of the listed prefixes (for example, "Dr" followed by punctuation does not signify a sentence split).[1] This approach was selected since it was not trained on any outside data and provides decent sentence breaks.

Additional preprocessing included removal of blank lines, and lower casing the entire "QED" dataset, which came in all capital letters.

3.2 Tokenizer

In order for the model to train on the data, a tokenizer must convert the input sentences into tokens. Word-level tokenizers replace any words not seen in the training data with an unknown token. Given the small amount of training data available in this challenge, this would result in many words marked as unknown. At the other extreme, character-level tokenization breaks every input into characters in order to eliminate any unknown tokens from occurring. This also has the advantage of having a small vocabulary size, since it consists only of characters. A major drawback of this approach is that, unlike words, characters may not have meaning by themselves. A popular and successful approach sits between these two by merging frequent pairs of characters together iteratively to create a vocabulary of characters and merged tokens. This approach is known as byte-pair encoding (BPE) (Sennrich et al., 2015). Despite its name, byte-pair encoding applied to natural language models typically does not operate at the byte level. A more recent approach used in language models such as GPT-2 (Radford et al., 2019) is byte-level byte-pair encoding. This is similar to earlier BPE, but operates directly on the byte representations and has been effective in language models.

After preprocessing, a byte-level byte-pair-encoding tokenizer was trained on the data. The vocabulary size was set to 52,000 with special tokens added for sentence beginning and end, padding, masking, and an unknown token in case any bytes were never seen in the training data. The maximum length was set to 128 (126+beginning and end tokens). Once trained, this tokenizer was used across models for consistency.

4 Model and Training

Our model is a RoBERTa (Liu et al., 2019) model. RoBERTa improves upon the BERT (Devlin et al., 2018) model, increasing performance across a range of benchmarks. While the architecture of both models is nearly identical, there are a number of smaller changes made in RoBERTa. Among the most relevant for his work is the removal of next sentence prediction task during pretraining and modifying the masked language mod-

[1] https://github.com/mediacloud/sentence-splitter

eling pretraining task by re-selecting the masks each training epoch. The architecture underlying these models is the Transformer model (Vaswani et al., 2017). The "base" model and the "CL-sm" model are the same size and number of parameters, differing only in how they were trained. The "CL-lrg" model is trained in the same way as "CL-sm" but is a slightly larger model. More details of the models are discussed in section 5.1. The "CL-sm" model trained for 5 epochs was submitted to the BabyLM Challenge[2]. The "CL-lrg" model trained for 10 epochs is also available for download[3]. Since we do not significantly modify this underlying architecture, we leave the details of these models to their respective papers. Code to train our model can be found on GitHub [4].

4.1 Masked Language Modeling

The pretraining objective used to train our models was masked language modeling. In masked language modeling (MLM), tokens are randomly replaced with a special ¡mask¿ token. Given the surrounding context, the model predicts the masked token and the loss is used to train the model. As mentioned above, MLM as a pretraining task for language modeling has been used successfully in many existing models such as BERT and RoBERTa. Following RoBERTa, masks were computed dynamically for each training instance and were not retained across epochs.

4.2 Curriculum Learning

Our models used curriculum learning to gradually increase the difficulty of the training set. As discussed earlier, there are ten datasets that were combined to create the training data. Each of these datasets were added one at a time to increase the training data. The way "difficulty" was measured, avoiding applying outside knowledge to the data, was by dividing each of the ten data files' size by the number of lines in that file. This gave an approximate bytes per line ranking of the ten training files. This was computed after all preprocessing was done, including the additional line splits and blank line removals.

A number of epochs is chosen prior to pretraining. After that number of epochs of training, another dataset was added to the training data. The model weights from the end of the previous epochs were used, but the learning rate and other hyperparameters were reset. As there was more data in the training set as training continued, the epochs contained more updates the further the training went. The final set of epochs included all of the training data.

The order in which datasets were added by following this approach was:

1. CHILDES (MacWhinney, 2000)
2. OpenSubtitles (Lison and Tiedemann, 2016)
3. Switchboard Dialog Act Corpus (Stolcke et al., 2000)
4. British National Corpus, dialogue portion[5]
5. QCRI Educational Domain Corpus (QED) (Abdelali et al., 2014)
6. Simple Wikipedia[6]
7. Children's Book Test (Hill et al., 2016)
8. Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2020)
9. Children's Stories Text Corpus[7]
10. Wikipedia[8]

This ordering also orders spoken, transcribed datasets before written datasets. This follows the language acquisition and exposure ordering that human children encounter. The exact ordering differs from an ordering based on when children would be exposed to these particular datasets, in particular Children's Stories Test Corpus would come much earlier in the order. One benefit of our approach is that it can be applied to any datasets without prior knowledge of what the datasets contain.

Unlike many other works which combine data from all sources into one pool before assigning an order to samples, this work places the ordering on the data sources themselves. This approach is fitting for settings such as this one in which the data from different sources can differ widely in their complexity. Datasets which contain more similar sources may not benefit from this approach, but that is outside the scope of this current work.

Since our tokenizer uses byte level byte pair encoding, we chose to explore a byte-based ranking for the dataset complexities.

5 Results

Here we examine the results of models on the provided evaluation benchmarks (Gao et al., 2021).

[2]https://huggingface.co/jdebene/BabyLM-jde-5/tree/main

[3]https://huggingface.co/jdebene/BabyLM-jde-larger-10/tree/main

[4]https://github.com/jdebened/BabyLM2023

Model	Ana. Agr.	Agr. Str.	Bind.	Ctrl. Rais.	D-N Agr.	Ellip.	Fill. Gap	Irreg.	Isl.	NPI	Quan.	S-V Agr.	Avg.
Base													
5 ep	72.65	66.59	64.84	60.96	85.12	51.39	63.60	90.69	34.19	57.52	78.77	57.85	65.35
10 ep	79.35	70.42	68.06	64.85	94.76	65.65	65.66	**92.32**	34.68	59.57	**79.01**	62.48	69.73
20 ep	84.25	72.65	68.60	65.11	96.55	70.44	68.04	91.09	32.21	55.86	72.23	67.37	70.37
CL-sm													
5 ep	81.65	72.77	**71.40**	67.34	96.38	71.94	68.32	81.63	33.48	65.87	69.22	71.29	70.94
10 ep	86.50	72.81	69.46	68.91	94.79	**75.87**	71.68	80.46	39.05	62.22	67.95	72.68	71.87
CL-lrg													
5 ep	84.92	**73.44**	70.36	**69.07**	**97.14**	74.31	74.07	85.70	34.87	64.14	74.91	72.86	72.98
10 ep	**87.88**	71.40	70.04	68.94	94.75	75.75	**74.56**	84.17	**44.25**	**67.86**	66.18	**77.76**	**73.63**

Table 2: Comparison of models on BLiMP tasks. Average shown is macro-average across all tasks. Models trained using curriculum learning surpassed baseline (all data, no curriculum learning) and improved further when more epochs were used for training. **Bolded** values show best in column.

5.1 BLiMP

Distributed as part of the BabyLM challenge was an evaluation pipeline. This pipeline included zero-shot evaluation on tasks from the BLiMP benchmark (Warstadt et al., 2020a). The BLiMP data was filtered to only include words which appeared at least twice in our training dataset (strict-small track) [9]. BLiMP (The Benchmark of Linguistic Minimal Pairs) provides a pretrained language model with a pair of sentences to score. The sentence pairs have small differences designed to assess whether a language model can select the correct sentence. If the language model assigns a higher score to the correct sentence in the pair, it is marked as correct. The tasks within BLiMP test different phenomena spanning syntax, semantics, and morphology. Since the sentences come in pairs, a random guessing baseline would achieve around 50% accuracy across all tasks.

Table 2 shows the results on BLiMP tasks. All models shown used the same preprocessing, tokenization, and are RoBERTa models. The base model had six attention heads and four hidden layers. All data was used for every epoch of training the base model. The "CL-sm" model also had six attention heads and four hidden layers, thus maintaining the same architecture. The curriculum learning technique described above was applied at training time, gradually increasing the amount of available training data. The "CL-lrg" model is a larger version with twelve attention heads and six hidden layers. The curriculum learning technique is the same as was used for the smaller model.

As can be seen in Table 2, even with the limited amount of training data available in this challenge, the language models were able to improve on most BLiMP tasks. The models trained using a curriculum learning approach all had higher average scores across the BLiMP tasks. The only two tasks in which the base model outperformed the curriculum learning models were irregular forms and quantifiers. The irregular forms task focuses on irregular forms of words in English for past particles. The example given in the BLiMP paper for the irregular forms task is: "Aaron broke the unicycle" compared to "Aaron broken the unicycle". For the quantifiers task, grammatical use of quantifiers is tested as shown in the example from the BLiMP paper: "No boy knew fewer than six guys" compared to "No boy knew at most six guys".

Upon further inspection of the training data, this drop in performance on the irregular forms makes sense given the order in which the curriculum learning datasets were used. Initially, the model trains exclusively on the CHILDES dataset. After the specified number of epochs, the OpenSubtitles data is added and additional training is done. As this process continues, the model appears to be heavily influenced by the improper use of irregular forms within the CHILDES dataset. For example, "you broken the trains ?" is a sentence in the dataset in which the speaker is likely repeating a statement made by the child. By contrast, the model is exposed to every dataset during every epoch in the base model. The training data coming from sources such as Wikipedia, simple Wikipedia, Project Gutenberg, and others is much less likely to feature many improper uses of irregular forms.

The performance drop on the quantifiers task is not as obvious in the data, nor is the drop in performance as dramatic. Even within the base model itself, performance on the quantifier task dropped when moving from 10 epochs of training to 20 epochs of training. Training models on larger portions of the datasets included in this challenge

[5] http://www.natcorp.ox.ac.uk

[6] https://dumps.wikimedia.org/simplewiki/20221201/

[7] https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus

[8] https://dumps.wikimedia.org/enwiki/20221220/

[9] See https://github.com/babylm/evaluation-pipeline for more details

may provide more insight into which datasets contribute positively or negatively toward each task in the benchmark. This is left to future work outside of this challenge.

Another task of note is the island effects task. This task assesses how well the language model learns that certain syntactic structures prevent syntactic dependencies across them. This phenomenon is investigated in works such as by Kush et al. (2018). An example of this, given in the BLiMP paper, is: "Whose hat should Tonya wear?" compared to "Whose should Tonya wear hat?". It is noted in the BLiMP paper that this is the hardest task in the benchmark for models they tested. Our models not only did not do better than random chance (50%), they actually consistently preferred the wrong option. Similar to quantifiers, there may be interesting results from uncovering why these models prefer the sentences which violate the island effects, but that is left to future work outside the scope of this challenge.

5.2 SuperGLUE

The GLUE benchmark (Wang et al., 2018) was designed to assess natural language systems on language understanding tasks. There were nine tasks aimed at testing different aspects of the language understanding problem. About a year after its release, in response to rapid improvements on the benchmark by natural language systems, SuperGLUE was published as a more challenging supplement or replacement (Wang et al., 2019).

Since these tasks require more than just a language model score to make predictions, the provided evaluation scripts finetuned a model for each task. The finetuning process involves a small amount of additional, task-specific training of a pretrained model in order to boost performance or add a suitable encoder or decoder layer for the specific task. The initial learning rate was set to 5e-5, the batch size set to 64, and the model trained for up to 10 epochs.

The results for tasks from GLUE and SuperGLUE can be seen in Table 3. The models trained with curriculum learning had higher average scores than those trained conventionally. While the curriculum learning models improved on most tasks, there were three tasks worth examining further: QNLI, BoolQ, and WSC.

The task labeled QNLI (Question-answering NLI) comes from the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). In SQuAD, systems were provided with a question and a paragraph which contained a sentence answering the question. The task was to pick out which sentence answered the given question. This was converted into the QNLI task by pairing the question with each sentence in the given paragraph and asking a natural language system to classify whether the answer to the question is contained in the given sentence.

In our results, we can see that the curriculum learning approach has the highest score of any of our models after its shortest training set of 5 epochs. However, performance dropped when the pretraining within the curriculum learning framework was increased to 10 epochs per set of data. Performance degraded even further when the model size was increased and the curriculum remained the same.

For the task labeled BoolQ (Boolean Questions) (Clark et al., 2019), the task is to provide a boolean response (yes/no) to a question. The system is provided with the question and a paragraph from a Wikipedia article which contains the answer to the question. Here we see a similar phenomenon to the trend with QNLI. The curriculum learning models' performance decreases when allowed more epochs for pretraining. Increasing model size had a less noticeable drop in performance.

The WSC (Winograd Schema Challenge) task (Levesque et al., 2012) requires a system to pick to which noun phrase in a sentence a pronoun is referring. The system is provided with a sentence which includes a pronoun and noun phrases. The pronoun refers to one of the noun phrases. The drop in performance for models which trained for more epochs is fairly consistent across models, regardless of whether curriculum learning was applied for pretraining or not.

Despite these three tasks, average performance across the benchmark does improve when using curriculum learning, when increasing the number of pretraining epochs, and when increasing the model size.

5.3 MSGS

The MSGS (Mixed Signals Generalization Set) (Warstadt et al., 2020b) was designed to test for inductive biases in pretrained language models. The aim of these tests are to not only find whether a language model represents certain phenomena, but more importantly whether it has learned to prefer them when generalizing. As was done for the SuperGLUE tasks, finetuning is done for each model to find its performance on each task. Our finetuning hyperparameter setup is unchanged for MSGS.

The results shown in Table 4 show that the performance across our models was relatively similar. The conventional training method used for the base model had nearly identical average performance across all three different training lengths with the exception of poor performance on the SC-LC task for the model trained for 20 epochs. Given the consistency across other tasks, it is possible that retraining would not replicate this drop, though

Model	CoLA	SST-2	MRPC (F1)	QQP (F1)	MNLI	MNLI-mm	QNLI	RTE	BoolQ	Multi RC	WSC	Avg
Base 5 ep	70.76	84.84	76.92	77.07	67.02	67.84	62.20	48.48	63.35	57.94	61.45	67.08
10 ep	71.05	85.63	74.05	77.30	67.65	69.67	62.64	44.44	63.07	50.82	59.04	65.94
20 ep	70.36	86.61	78.63	77.77	68.07	69.37	65.27	44.44	65.70	58.38	59.04	67.60
CL-sm 5 ep	71.34	84.84	73.90	77.69	65.79	66.52	66.54	46.46	67.36	59.04	61.45	67.36
10 ep	72.33	87.99	76.45	78.47	70.05	71.23	64.22	45.45	64.73	59.58	56.63	67.92
CL-lrg 5 ep	72.33	87.01	79.38	78.60	70.71	72.15	63.87	47.47	65.42	57.28	61.45	68.70
10 ep	74.39	88.19	79.41	78.57	70.05	70.56	63.17	51.52	64.87	59.58	59.04	69.03

Table 3: Comparison of models on (super) GLUE tasks. Average shown is macro-average across all tasks. Models trained using curriculum learning surpassed baseline in average performance.

Model	CR-ctrl	LC-ctrl	MV-ctrl	RP-ctrl	SC-ctrl	CR-LC	CR-RTP	MV-LC	MV-RTP	SC-LC	SC-RP	Avg
Base 5 ep	82.13	100	97.76	99.29	95.25	66.46	66.64	66.61	66.38	88.69	69.75	81.72
10 ep	84.36	100	97.77	98.64	93.46	69.11	66.81	66.61	66.72	89.53	65.07	81.64
20 ep	89.94	100	97.98	99.98	89.92	66.60	66.92	66.61	66.79	67.39	64.56	79.70
CL-sm 5 ep	91.14	100	97.45	99.74	86.71	66.49	67.15	66.61	66.87	63.84	62.34	78.94
10 ep	88.37	100	97.93	100	89.96	66.38	67.29	66.61	66.78	70.10	65.72	79.92
CL-lrg 5 ep	84.57	100	99.36	98.94	94.39	66.35	67.01	66.61	66.62	72.69	70.33	80.62
10 ep	89.56	100	99.87	100	92.21	67.00	66.76	66.61	66.65	75.54	69.30	81.22

Table 4: Comparison of models on MSGS benchmark tasks. Average shown is macro-average across all tasks. Models trained using curriculum learning performed slightly worse than baseline model, but improved with more epochs. The base model, by contrast, had worse performance with more training epochs.

that would need to be tested to be confirmed. The models trained with curriculum learning had slight improvements when pretrained for more epochs as well as when the model size was larger. Overall, the techniques used in this work showed little impact on the MSGS tasks.

6 Conclusion

Large language models have been highly successful across a wide variety of tasks in Natural Language Processing. Due to the rapidly increasing model size and training data size, however, the cost to train new models is prohibitively expensive for many researchers. The BabyLM Challenge is a shared task designed to highlight methods for training language models at a smaller scale. These methods may lead to improvements in scaling up training more efficiently, training language models in low-resource settings, and drawing upon the way human children acquire language.

In this work, the strict-small track allowed our models to use a given dataset containing around ten million words from data sources that a child may encounter when learning language. No tools which used outside data for pretraining were allowed, reducing the ability to use many existing pipelines. This restriction is realistic for many low-resource scenarios in which these tools are lacking.

This work explores ordering training data by bytes per line for a curriculum learning approach. This measure of difficulty is inspired by the use of byte-based byte-pair-encoding tokenization and is easy to apply without needing any domain knowledge of the dataset. The results show that curriculum learning with this setup obtains improved results on benchmark evaluations when training for a set number of epochs. In settings in which additional tools, data, or computational resources are available, this curriculum setup is easy to apply and further evaluation in those settings is a potential area for future work.

This work used the Augie High-Performance Computing cluster, funded by award NSF 2018933, at Villanova University.

Limtations

This work was completed as part of the BabyLM Challenge. As such, additional testing would be required to determine how well the results generalize outside of this data setting. In a similar way, pretraining settings in which some pre-existing tools which are trained on outside data are available may produce different results. Additionally, if more computational resources are available, the benefit to the models when trained for more epochs remains to be seen. Other work on curriculum learning found faster convergence, but models in this work were trained for a set number of epochs and not to convergence. The results outperform the baseline model at the set number of epochs used, but training to convergence may lead to better or worse results.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In *LREC*, volume 14, pages 1044–1054.

Hadi Amiri, Timothy Miller, and Guergana Savova. 2017. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410.

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, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. arxiv 2015. *arXiv preprint arXiv:1511.02301*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Dave Kush, Terje Lohndal, and Jon Sprouse. 2018. Investigating variation in island effects: A case study of norwegian wh-extraction. *Natural language & linguistic theory*, 36:743–779.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

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

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

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.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2023. Length-based curriculum learning for efficient pre-training of language models. *New Generation Computing*, 41(1):109–134.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020a. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.