# Creating Masked Language Model From Human Nature Imitation Perspective

**Fatih Erdoğan**
ferdogan20@ku.edu.tr

## 1 Problem

NLP researchers have developed various model architectures, among which the transformer model has demonstrated state-of-the-art results. The advancement in computational power has allowed developers to train models using vast amounts of data and larger parameter sizes, leading to improved performance. It is important to acknowledge that the progress achieved so far is a combination of the imitation of human nature through neural networks and contribution of the available computational resources. However, considering the limited amount of data that humans are exposed to compared to what current language models are trained on, it becomes evident that existing training methods cannot get the most out of data. Therefore, there is a need for advancements in training methods that can leverage the vast amount of available data. This would not only aid in constructing better language models for low-resource languages but also enable even more remarkable results with reduced computational power and data requirements. Motivated by this objective, this project (inspired by the BabyLM challenge [1]) aims to train a language model from scratch. The goal is to outperform baseline models trained using traditional approaches, while utilizing the same amount of data that a 13-year-old child would typically be exposed to. By doing so, this project seeks to demonstrate the potential of more efficient and effective training methods.

## 2 Proposal vs. Accomplishments

As mentioned in the proposal, I have developed a language model from scratch using curriculum learning (CL). During training, as the complexity of the data was increasing, I increased the parameter size of the model and added new self-attention layers. Here is a summary of the proposed methods before:

- ~~Integration of additional self-attention layers into the transformer language model~~
- ~~Utilization of curriculum learning~~ *including the adjustment of the learning rate based on the complexity of the data*: There was a problem related to the converge of the loss. Therefore, I used a learning rate scheduler and couldn't adjust the learning rate.
- ~~Initially, training a model with lower number of parameters using less complex data, followed by the reconstruction of another transformer with increased parameters which are initialized with the parameters of the previous model~~
- *Creating a model outperforming the baseline*: Due to time constraints and the limitations of training a model on Google Colab, I was unable to generate a model that surpasses the baseline performance

## 3 Related Work

To increase the performance and decrease the amount of time required for training models, various CL methods are proposed by prior studies. A good proportion of the CL related research was focused on the development of data complexity definitions to organize a curriculum and the finetuning of the models for various downstream tasks such as NLU and NMT (Xu et al., 2020; Zhang et al., 2018; Platanios et al., 2019; Zhang et al., 2019).

Based on their experiments on the sentiment analysis task with LSTM's, Cirik et al. (2016) proposed that training with CL increases the performance especially when training smaller models.

[1] link for the project description https://arxiv.org/pdf/2301.11796.pdf!

In the context of language modeling, most of the prior research provided evidence for the benefits of CL which makes this topic worth investigating.

Campos (2021) used CL to train ELMo which is a Bi-LSTM model from scratch using the original parameters. Based on the evaluation of the model on the perplexity on the training corpus and the transfer ability on GLUE tasks, he concluded that the CL has no significant effect on the performance of the model. On the other hand, Nagatsuka et al. (2021) conducted experiments on CL with transformer architecture and discovered significant improvements. Defining the data complexity as the sequence length, they investigated the effects of gradual increase in the block size during training on the convergence speed and the model performance. They trained a RoBERTa model using the parameters in the original paper. To compare the model performance, they used RoBERTa trained with random sampling as the baseline. The results revealed that the CL approach used in this research decreased the convergence speed by approximately 20% and it outperformed the baseline model in 6 different GLUE tasks. From a different perspective, Lee et al. (2022) measured the "easiness" with the number of conceptual relations that a word has with its neighbors and Wang et al. (2023) organised a curriculum starting from the most frequent word.

My approach to CL diverges from the mentioned research in how data complexity is defined, specifically regarding the exposure of the data throughout an individual's lifespan. And most importantly, the idea of increasing the model parameters along the training period is introduced which does not exist in any of the previous studies to my knowledge.

## 4 Dataset

In the project, I used the Strict-Small dataset provided by BabyLM team. It is composed of 10 files from various domains such as child-directed speech, dialog, children books, subtitles, and Wikipedia. The total word count contained in the dataset is 10M. You can find the details of the dataset in Table 1. I used the first 2 for phase 1, following 3 for phase 2 and the last 5 for the phase 3.

Data preparation is one of the important challenges for training a model with curriculum learning as in addition to data collection, it requires a method to organize the data for the learning phases. However, as the data prepared by the BabyLM was already grouped into ten files, I reordered them according to the exposition of these data in human life. For example, in the first phase of the training, I included the Childes dataset which was composed of child-directed speech such as: "oh my how did you feel ?". I used the files which include longer and more complex sentences such as Wikipedia data at the last phase.

| | Dataset | Word Count |
|---|---|---|
| **1** | Childes | 0.44M |
| **2** | QED Corpus | 1.04M |
| **3** | OpenSubtitles | 3.09M |
| **4** | Switchboard | 0.12M |
| **5** | Children's Book Text | 0.57M |
| **6** | Children Stories | 0.34M |
| **7** | Gutenberg Corpus | 0.99M |
| **8** | Simple Wikipedia | 1.52M |
| **9** | Wikipedia | 0.99M |
| **10** | BNC Dialogue | 0.86M |
| **Total** | | 9.96M |

Table 1: Training dataset for strict-small task from BabyLM team (Warstadt et al., 2023)

### 4.1 Data Preprocessing

Even though I tried to truncate the sentences during the training, some of the sentences exceeding the maximum token size of the model caused error. Thus, I split the lines whose tokenized version was exceeding the maximum length of a sequence that RoBERTa can handle (512 tokens), into more lines and saved that one as a different file.

## 5 Baselines

There are three baseline models provided by the BabyLM Team (Warstadt et al., 2023) which are initialized with the hyperparameters from established large language models. They used hyperparameters from OPT (Zhang et al., 2022), RoBERTa (Liu et al., 2019)and T5 (Raffel et al., 2020). RoBERTa-base being a transformer based encoder is trained on the masked language modeling task. T5-base is a transformer based encoder-decoder and trained on text-to-text generation. The third baseline OPT-125M is a decoder-only model which is trained on causal language modeling. The training of all three models utilizes the data specified in section 4.

| Model | A.A. | A.S. | Bin. | C.R. | DN A. | Ell. | F.G. | I. F. | I. E. | NPI L. | Qu. | SV A. | # P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPT | 63.8 | 70.6 | 67.1 | 66.5 | 78.5 | 62 | 63.8 | 67.5 | 48.6 | 46.7 | 59.6 | 56.9 | 125M |
| RoBERTa | 81.5 | 67.1 | 67.3 | 67.9 | 90.8 | 76.4 | 63.5 | 87.4 | 39.9 | 55.9 | 70.5 | 65.4 | 123M |
| T5 | 68.9 | 63.8 | 60.4 | 60.9 | 72.2 | 34.4 | 48.2 | 77.6 | 45.6 | 47.8 | 61.2 | 65.0 | 220M |
| Phase 1 | 39 | 55.5 | 58.5 | 57.6 | 50.3 | 41.7 | 66.6 | 36.6 | 45 | 36.6 | 38.82 | 49.5 | 2.5M |
| Phase 2 | 46.2 | 57.4 | 63.8 | 57.4 | 54.2 | 33.8 | 64.8 | 70.1 | 33.1 | 59 | 37.5 | 50.9 | 16.7M |
| Phase 3 | 49.3 | 60.3 | 63.4 | 58.3 | 65.2 | 34.4 | 62.5 | 81.7 | 47.7 | 53.7 | 59.7 | 51.31 | 109M |

Table 2: BLiMP accuracy scores of baselines and my models with number of parameters displayed at the rightmost colum. A.A.: Anaphor Agreement, A.S.: Argument Structure, Bin.: Binding, C.R.:Control/Raising, DN A.: Determiner Noun Agreement, Ell.: Ellipsis, F.G.:Filler-Gap, I.F.: Irregular Forms, I.E.: Island Effect, NPI L.: NPI Licensing, Qu.: Quantifiers, SV A.: Subject-Verb Agreement, # P: Number of parameters

| Model | CoLA | SST-2 | MRPC (F1) | QQP (F1) | MNLI | MNLI-mm | QNLI | RTE | BoolQ | MultiRC | WSC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OPT | 64.6 | 81.9 | 72.5 | 60.4 | 57.6 | 60 | 61.5 | 60 | 63.3 | 55.2 | 60.2 |
| RoBERTa | 70.8 | 87 | 79.2 | 73.7 | 73.2 | 74 | 77 | 61.6 | 66.3 | 61.4 | 61.4 |
| T5 | 61.2 | 78.1 | 80.5 | 66.2 | 48 | 50.3 | 62.0 | 49.4 | 66.0 | 47.1 | 61.4 |
| Phase 1 | 64.8 | 77.5 | 81.1 | 61.1 | 46.4 | 48.5 | 60.8 | 55.6 | 60.4 | 55.4 | 61.4 |
| Phase 2 | 63.6 | 78.1 | 75 | 63.6 | 51.7 | 53.3 | 59.8 | 41.4 | 63.1 | 56.2 | 61.4 |
| Phase 3 | 66.5 | 79.3 | 76.6 | 58.4 | 52.9 | 54 | 58.3 | 50.5 | 63.3 | 53.9 | 61.4 |

Table 3: Super GLUE accuracy (unless otherwise noted by (F1)) scores of baselines and my models.

The evaluation metrics and the code for evaluating the models are provided by the BabyLM Team. The evaluation metrics involve 2 methods on multiple metrics. First one is a zero-shot evaluation on the Benchmark of Linguistic Minimal Pairs (BLiMP) and the second one involves the fine tuning of the model for General Language Understanding Evaluation (GLUE). Table 2 and Table 3 displays the scores from the evaluation metrics on the test set for both the baseline models and the models trained with the proposed strategy. Unfortunately, it is clearly seen that the model trained with the approach introduced in section 6 cannot compete with the baseline models.

## 6 My Approach

Inspiring from the human learning nature, I prepared a 3 phase learning curriculum for the masked language modeling task. As the best performing model among the baselines is RoBERTa, I used it in my experiments to test my approach for the problem.

Human brain is developing gradually. New connections are established between the neurons and the perception of already acquired knowledge is subject to change. Therefore, one can conclude that the learning process of humans involves first capturing the essence of the subject and then delving into its deeper intricacies and complexities. Having the motivation from this natural learning process, I developed three RoBERTa model for each learning phase with different parameter sizes. From the HuggingFace transformers library I used RobertaForMaskedLM as the template model and used ByteLevelBPETokenizer from tokenizers library to create a customized tokenizer. You can find the model and training related hyperparameters at Table 4.

| | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| **voc_size** | 5334 | 15334 | 30334 |
| **# of layers** | 3 | 6 | 12 |
| **hidden size** | 192 | 384 | 768 |
| **FFN in. hid. s.** | 768 | 1536 | 3072 |
| **# of self-att_h** | 3 | 6 | 12 |
| **att. head size** | 64 | 64 | 64 |
| **dropuput** | 0.1 | 0.1 | 0.1 |
| **att. dropout** | 0.1 | 0.1 | 0.1 |
| **batch size** | 64 | 64 | 32 |
| **epochs** | 10 | 5 | 5 |
| **max seq. len.** | 512 | 512 | 512 |
| **acc. steps** | 4 | 4 | 4 |
| **mlm prob.** | 0.15 | 0.15 | 0.15 |
| **optimizer** | AdamW | AdamW | AdamW |
| **scheduler** | lin. | lin. | lin. |
| **max lr.** | 5e-5 | 5e-5 | 5e-5 |

Table 4: model hyperparameters (RoBERTa) for each phase of the curriculum

The reason behind the choice of byte level BPE tokenization is that it is more intuitive to initially learn the most common character combinations appearing together and also the original RoBERTa

model (Liu et al., 2019) used this tokenization. Due to the compatibility issues, I converted this customised tokenizer to RobertaTokenizer. During the transition between the phases, the parameters of the previous model is copied to related positions at the subsequent model. Figure 1 illustrates the general idea used for initializing the parameters of the following model.

This method is applied for each one of the embedding layer, encoder layers (for the ones that existed in the previous model), and language model head layer parameters. Differing from the others parameter sets, I used a special initialization technique for the word embeddings and the decoder of the language model head layer. The idea behind this method is that the token which is obtained from the combination of two subtokens would carry similar information to both of the tokens that it is constructed from. This may be not useful for the first some number of merges as they are likely to carry generic information in their embeddings, but after a number of merges, it is likely that the embeddings of the subtokens can be a good starting point for the new tokens.

To achieve the mentioned operations and to have a consistency between the previous and subsequent models, the construction of the tokenizer plays a significant role. For a new phase, the new model is created with an extended vocabulary. The token indices of the previous model must remain the same for the next model so that the initialization method that I mentioned make sense. To achieve that, I applied the byte level BPE to the same training corpus which is composed of all training files.

I increased the number of self-attention heads proportional to the hidden size (word embedding size). The idea behind this choice is again ensuring the compatibility between the previous and next model. During the training of a model, each attention head focuses on a specific part of the word embeddings. Therefore by increasing the number of heads proportional to the embedding size, I aimed to keep the attention head size constant between the phases. Thus, even though the results of each head is affected from the dimensional increase in the input, at least the portion of the input vector inherited from the previous phase interacts with the attention parameters of it.

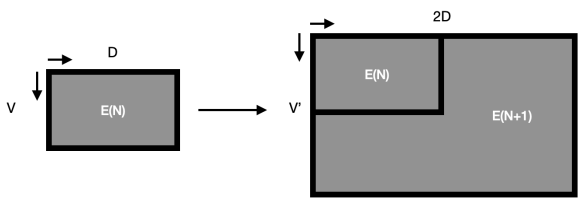


Figure 1: Initialization of the word embedding matrix for the subsequent model

I managed to complete a working implementation. However, I do not consider the project as a complete one. Because due to time constraints and the lack of computational resources to train the models, I was able to perform only 10, 5 and 5 epochs of training for 1st, 2nd and 3rd phase respectively. In addition, again because of the time constraints, I couldn't tuned the hyperparameters for training.

The idea and the code related to the initialization of the model using the previous completely belongs to me. I checked the documentation of transformers[2] and tokenizers[3][4] library from HuggingFace. For the training set up, I revised the assignment codes provided during the semester and also checked the example codes[5] on a public repository for transformers library.

I worked on Google Colab during the experiment. However, I encountered significant challenges that were quite disagreeable. The disconnection problem costed me a significant amount of time. In addition, from time to time, as I was using a free account, I wasn't able to access the GPU resources of the system which obligated me to purchase the paid version. Even after this, the disconnection problem wasn't solved completely. After several painful experiences, I inserted a line of code which saves the model after each epoch of training to minimize the time loss from a possible disconnection. Additionally, to solve memory issues related to GPU, I needed to decrease the batch size for training the model on third phase.

[2]AutoModelForMaskedLM - HuggingFace https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModelForMaskedLM!

[3]RobertaTokenizer - HuggingFace https://huggingface.co/docs/transformers/model_doc/roberta#transformers.RobertaTokenizer!

[4]RobertaTokenizer - HuggingFace https://github.com/huggingface/tokenizers/blob/main/bindings/python/py_src/tokenizers/implementations/byte_level_bpe.py!

[5]RobertaTokenizer - HuggingFace https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_mlm.py!

## 7 Error Analysis

Considering the zero-shot evaluation scores on BLiMP metrics, it is seen that the final model resulting from phase 3 failed to outperform the baseline models. Although there may exist multiple reasons behind this failure, I believe that one of the most important ones is the limited epochs of training.

The most eye-catching result concerning final model is Irregular Forms which is probably related to the vocabulary size of the model. Compared to the other metrics, the model performed better at D-N Agreement and Filler-Gap tasks, which can be interpreted as model having an understanding of syntactic relations. Nevertheless, its poor performance on Ellipsis and Anaphor Agreement reveals its incapability of distinguishing semantic relations.

## 8 Conclusion

I first started studying NLP during this semester. Therefore in addition to understanding the concepts and model architectures, I needed to get myself familiarized with the libraries providing NLP frameworks. The modules are surprisingly incompatible forcing the user to repeat some steps during the development. Moreover, I also needed to adapt myself to a more rigorous working style as trial-and-error method has a huge time cost. Another valuable contribution of this project to me was it required me to read a lot of paper which gradually increased my understanding on the subject.

Even though the result I obtained weren't satisfying, I still think that the curriculum learning is an important subject to investigate. The study presented in this paper can be improved through the development of a customized model, whose inherited parameters can be either frozen or set to an exceptionally low learning rate. Also, this idea can be combined with a proposed curriculum from the previous research which is proven to be showing better results.

## 9 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, I used GPT3.5 to construct the tables and sometimes for sentence correction.

*If you answered yes to the above question, please complete the following as well:*

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - I have this line in latex but I want the "datasets ... performance" part not be stroke out: ¡latex expression¿
  - Can you prepare me a table composed of 4 columns and 12 rows where the last row is separated from the otherwith a line
  - I want this table to be placed at the bottom the page as a single column while the rest of the paper is double column
  - but this table appear at another page
  - I want no lines between rows
  - for toprule and bottomrule it says undefined control rule
  - still same issue
  - make the previous table constructed as this table: ¡table¿
  - ...
  - can you improve this: I meet NLP in this semester. Ans: This semester, I had the opportunity to delve into the fascinating world of NLP.
  - more formal Ans: During the course of this semester, I had the privilege of engaging with the discipline of Natural Language Processing (NLP).
  - this semester is the first time I worked on NLP Ans: This semester marked my inaugural exploration into the realm of Natural Language Processing (NLP).
  - bro make this more normal what are the words you are using Ans: I first started studying Natural Language Processing (NLP) during this semester. (used at the first line of conclusion)
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant?

Did you use it to generate new text, check your own ideas, or rewrite text?

- When I used it for table generation task, even though it struggled in some cases, after a few trial, it managed to generate the correct response. However, considering my previous experience as well, when I provide a very small task such as a sentence correction, it tries to generate a piece of art. When a longer sequence is provided, it generates more acceptable results. But if there is a vague statement which can be a sentence with a meaning gap, it tries to fill that with its own ideas.

## Limitations

The main focus of this project was the construction of an architecture expanding the model parameters with the data complexity. Even though the application of this idea to the word embeddings and to the decoder of the language model head seems intuitive, for the encoder layers, this may not be a good initialization point. For instance, when a model with 6 layers and one with 12 layers are compared, the responsibility assigned to the $5^{\text{th}}$ layer may not be the same in both models. Additionally the idea of initializing the word embeddings of new tokens from their subtokens significantly relies on the usage of byte level BPE algorithm. Therefore, for the languages that do not use spaces to separate words, this approach may not be very useful.

## References


Campos, D. (2021). Curriculum learning for language modeling. *CoRR*, abs/2108.02170.

Cirik, V., Hovy, E. H., and Morency, L. (2016). Visualizing and understanding curriculum learning for long short-term memory networks. *CoRR*, abs/1611.06204.

Lee, M., Park, J.-H., Kim, J., Kim, K.-M., and Lee, S. (2022). Efficient pre-training of masked language model via concept-based curriculum masking.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Platanios, E. A., Stretcu, O., Neubig, G., Póczos, B., and Mitchell, T. M. (2019). Competence-based curriculum learning for neural machine translation. *CoRR*, abs/1903.09848.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Wang, Y., Zhang, Y., Li, P., and Liu, Y. (2023). Language model pre-training with linguistically motivated curriculum learning.

Warstadt, A., Choshen, L., Mueller, A., Williams, A., Wilcox, E., and Zhuang, C. (2023). Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus.

Xu, B., Zhang, L., Mao, Z., Wang, Q., Xie, H., and Zhang, Y. (2020). Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models.

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K., and Carpuat, M. (2018). An empirical exploration of curriculum learning for neural machine translation. *CoRR*, abs/1811.00739.

Zhang, X., Shapiro, P., Kumar, G., McNamee, P., Carpuat, M., and Duh, K. (2019). Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.