Curriculum Learning for BabyLM

Saim Ishtiaq si403
Emma Rafkin epr41
Ismail Shaheen ias68
Muxiang Wen mw1463

Abstract

This paper explores the use of curriculum learning for training language models with extreme resource constraints. Using the BabyLM 10M text-only dataset we investigate whether training a transformer-based masked language model on increasingly complex data can improve overall performance. To that end, we define and compare four curriculum learning strategies based on different metrics of data difficulty: surprisal, syntactic complexity, concreteness, and Maximize Minimal Means. We evaluate the results of our models using BLiMP and EWoK benchmarks, which assess syntactic and semantic understanding, respectively. The curricula created using syntactic methods performed the best on BLiMP while those created using semantic methods performed the best on EWoK. Our findings reveal that curriculum learning can in fact significantly improve model performance under resource-limited conditions, especially if the choice of curriculum is aligned with the task of the final trained model.[1]

1 Introduction

Large Language Models (LLMs) have achieved impressive results across a variety of natural language understanding (NLU) tasks, but their success is often contingent on access to massive amounts of training data. The BabyLM challenge (Charpentier et al., 2025) poses an alternative vision: could we train high-performing language models using only the amount of data that a child is exposed to? The strictest version of the challenge, BabyLM 10M, limits training to just 10 million tokens of child-directed and story-like text, which forces participants to adopt to more sample-efficient training strategies.

The BabyLM challenge calls for strategies based on human development and cognition. In order to "teach" a model language with the same amount of data that a human experiences, it follows that one should use a method that mimics that of a child learning. In this paper, we investigate whether curriculum learning, a method for slowly increasing the difficulty of the content that a model is exposed to throughout training, can improve the performance of transformer-based masked language models under the BabyLM 10M constraint. We train BabyBERTA (Huebner et al., 2021), a lightweight transformer model designed for low-resource settings, using four curriculum strategies based on alternative definitions of textual difficulty: surprisal, syntactic complexity, concreteness, and maximal minimal means (MMM). While human development and model development are drastically different processes, taking into account how humans pick up on patterns in language could point us to methodology that can enhance model efficiency when under extreme data constraints as is the case in the BabyLM challenge.

2 Background

Humans learn information by starting with the easiest concepts and slowing increasing the difficulty of those concepts over time. Curriculum Learning (CL) (Bengio et al., 2009) was proposed to train models in a way that mimics this behavior. As displayed in Figure 1, the data is split up based on the difficulty of its content. The model is then trained in multiple stages: first solely on the easiest data, then on the easiest data as well the medium-difficulty data, and finally on all of the data. It has been shown that CL allows a model to converge faster and generalize more across a wide range of NLU tasks (Xu et al., 2020). CL relies on some sort of difficulty measurement for each input in the data. Differing methodologies can lead to drastically different data ordering and therefore different curricula. We outline four different methodologies for data difficulty measurement.

[1] Code for this paper can be found here:https://github.com/erafkin/enlp_final_curriculum_learning

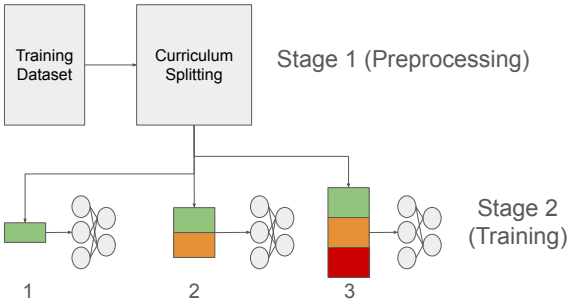

Figure 1: Process for training a model using curriculum learning. The training data is split into easy (green), medium (orange), and hard (red) data splits and the model is trained on the data in stages, each stage increasing the amount of data as well as the difficulty of the content.

2.1 Surprisal

Surprisal theory proposes that the amount of information contained in each word can be quantified using surprisal. Surprisal measures the predictability of a word $\mathcal{W}$ given its context $\mathcal{C}$: $-log(P(\mathcal{W} = w|\mathcal{C} = c))$ (Hale, 2001; Levy, 2008). It has been found that surprisal is correlated with word frequency and reading times, implying that the higher the surprisal of a word, the harder it is to process (Gibson et al., 2019). This correlation between human processing and surprisal motivates the use of surprisal for dividing curricula for language learning in the BabyLM challenge. Surprisal has been used for both ranking training sample difficulty (Ghebrechristos and Alaghband, 2019) and the BabyLM challenge before. For the 2023 BabyLM challenge, Hong et al. create an Active Curriculum Learning (ACL) model requiring an “oracle" to rank the data, for which they use surprisal. Their experiment, however, focused on the utility of ACL for the BabyLM task rather than purely testing surprisal as a metric for content difficulty. Surprisal calculations rely on word probability counts. This has historically been calculated with an N-gram model (Piantadosi et al., 2011), while more modern methods language models trained on vast corpora to obtain robust global probabilities (Salazar et al., 2020).

2.2 Syntactic Complexity

Syntactic Complexity ranks sentences based on how complex their grammatical structure is. It’s mainly used to convert simple sentences into their linguistically complex counterpart (Elgaar and Amiri, 2023). For CL, data can be sorted using a complexity score so that simpler sentences are used earlier in training and more complex sentences appear later in training. Syntactic complexity can be measured in a number of ways. One method calculates the syntactic weight of constituents as the difference in length in words between said constituents. Another method counts nodes to determine the number of nodes dominated (the more phrasal nodes a unit dominates, the more complex it is), while a third counts linguistic tokens that can be considered telltale signs of increased grammatical subordinateness and embeddedness, such as subordinating conjunctions, WH-pronouns, verb forms, and noun phrases (Szmrecsanyi, 2004).

2.3 Concreteness

Concreteness–the extent to which a word refers to something perceptible by the senses–has long been known to influence many aspects of language processing from word recognition speeds to memory recall. Prior concreteness rating systems (Paivio et al., 1968) covered only a few thousand words, limiting the applicability of concreteness as a variable in large-scale corpus or neuroimaging studies. In a 2014 study (Brysbaert et al., 2014), the researchers empirically gathered "concreteness" ratings for English words, filling a gap in psycholinguistic resources by covering tens of thousands of lemmas rather than the few hundred-few thousand found in earlier norms. The study provides one of the most comprehensive publicly available concreteness databases for English.

2.4 Maximize Minimal Means (MMM)

Curriculum learning, as seen from the above 3 strategies, structures training data to follow a progression from simpler to more complex examples. Maximize Minimal Means (MMM), proposed by (Salhan et al., 2024), extends this idea by incorporating fine-grained control over the training signal itself. Rather than staging full examples, MMM adjusts the masking probability per token based on the rarity of its syntactic (e.g., POS) or semantic (e.g., semantic role) tag. Tokens associated with frequent tags are masked more often, encouraging the model to learn basic linguistic structures early on. In contrast, tokens with rare tags are masked less frequently at the start, delaying their learning until the model has acquired stronger foundational representations.

This approach is grounded in psycholinguistic theories of language acquisition, where learners

are thought to internalize common grammatical patterns before rarer constructions. MMM thus reflects a cognitively plausible learning trajectory: it allows the model to focus on frequent, simpler patterns before gradually shifting attention to more complex ones. This is particularly beneficial for small-scale models with limited capacity, where premature exposure to rare phenomena can hinder generalization.

3 Methodology

We train a Masked Language Model (MLM) on curricula formulated through different difficulty ranking methodologies. We use the BabyBERTa (Huebner et al., 2021) architecture, which is the same as RoBERTa (Liu et al., 2019) but with only 8 layers, 8 attention heads, and with a context window of 128. The intended purpose of BabyBERTa was to create a transformer model that had similar performance to the state of the art but required less training data for low resource situations. This makes BabyBERTa ideal for the BabyLM challenge. We train each model for a total of nine epochs (three epochs per curriculum) on the BabyLM 2024 training data (Charpentier et al., 2025). We trained BabyBERTa models from initialization on curricula created using the following four methodologies that were introduced in Section 2. Surprisal, Syntactic Complexity, and Concreteness Ratings assign each input a "score" and then divide up the corpus into three curricula as described in Figure 1. The model trained using the MMM methodology is trained in phases using different masking probabilities.

3.1 Surprisal

In previous surprisal theory studies, a trigram model was deemed the most consistent (Piantadosi et al., 2011) N-gram size for predicting word probabilities in large corpora, therefore a trigram model was calculated from the BabyLM training corpus. An N-gram model was chosen over using probabilities from a pretrained language model due to time and resource constraints, however it also provides a localized understanding of sentence difficulty relative to the rest of the corpus which is what is needed for this task. While surprisal is generally calculated for individual words (or tokens), this task requires ranking input sentences or utterances by their information content. Therefore we had to calculate surprisal for sequences. This is generally done by summing the surprisal for each token in the series (Salazar et al., 2020). This method penalizes length, meaning that the easy curriculum would be primarily comprised of shorter sentences. Although surprisal does theoretically account for semantic plausibility we consider this method to be one that focuses on syntactic understanding rather than semantic understanding due the limited semantic information that can be gleaned from the N-gram model.

3.2 Syntactic Complexity

Previous work done on syntactic complexity has shown that SpaCy, a popular natural language processing toolkit of Python, was the most reliable tool to use due to its speed and scalability, robust dependency parsing, and overall ease of use (Spring and Johnson, 2022). We developed a weighted formula that combines four key linguistic features, each given a different weight, w: dependency tree depth ($w = 0.3$), clause count ($w = 0.4$), averaged word length ($w = 0.1$) and sentence length ($w = 0.2$). The weights were manually calibrated to reflect linguistic importance, redundancy between features, and diversity of signal. Structural features like clause count and dependency tree depth were prioritized over surface-level features such as sentence length and average word length to represent deeper syntactic complexity while avoiding overemphasizing verbosity or vocabulary alone. The dependency tree depth represented the maximum distance between a token and its syntactic head, whereas the clause count was identified by searching for specific dependency types like complement clauses, adverbial clauses, and relative clauses.

3.3 Concreteness

The Brysbaert concreteness rating study (Brysbaert et al., 2014) collected concreteness scores for about 40 thousand generally known English word lemmas. This study assigned each word with a concreteness score, ranging from a minimum of 1, representing something you cannot directly experience through actions or senses, to a maximum of 5, indicating something that you can directly experience in reality by actions like smelling, touching, hearing, and seeing–objects which can be physically pointed out. The BabyLM training data set consists of sentences rather than individual words. Using the concreteness ratings from this study, we scored the inputs of the BabyLM training dataset by averaging the concreteness ratings of each word in the input. Out of vocabulary (OOV) words did not

contribute to the score.

3.4 Maximize Minimal Means (MMM)

We adopt the MMM curriculum strategy to guide MLM training by varying masking probabilities based on linguistic tags as shown in Figure 2. Unlike staged curricula that reorder or filter training sentences, our approach retains the original data order and instead focuses on customizing the masking probability at the token level.

Linguistic Tagging: Each training sentence is annotated using the spaCy library. We apply both part-of-speech (POS) tagging and semantic tagging using the integrated PyMUSAS component, which provides fine-grained semantic class labels for each token. This dual annotation enables us to target subsets of tokens corresponding to specific syntactic or semantic roles.

Curriculum Masking Strategy: We define curriculum stages by selecting a set of target tags (e.g., NOUN and VERB for the initial stage) and assigning higher masking probabilities to those tokens. Specifically, for a given stage:

- Tokens with target tags are masked with a probability of 0.4
- All other tokens are masked with a baseline probability of 0.15

This simple yet effective mechanism allows us to focus the model's learning on desired linguistic features in a controlled manner. As training progresses, the set of high-probability tags is changed to include additional categories (e.g., adjectives, adverbs, or semantically marked tokens), guiding the model through a linguistically meaningful curriculum. Masking is applied dynamically during training, replacing the standard uniform 15% masking rate in MLM objectives with our tag-aware masking scheme.

4 Results

We compare the four models to two baseline models: one trained on curricula created randomly and one trained on the full dataset for each epoch. As is standard in the BabyLM challenge, we compare each model using EWoK and BLiMP (Ivanova et al., 2024; Warstadt et al., 2020). The EWoK dataset focuses on tasks with intangible, physical, semantic meaning. BLiMP generally targets syntactic understanding, but we note that BLiMP is separated into BLiMP and BLiMP Supplement. The

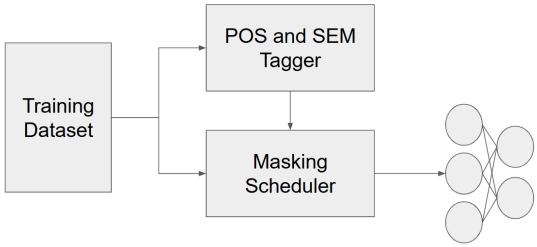

Figure 2: Process for training a model using MMM. The training data is tagged and the masking probability is changed according to the training stage and the tag associated with the token.

latter incorporates more semantic understanding tasks such as "blimp_supplement_hypernym" and "blimp_supplement_turn_taking". We expected the methods using semantics (MMM and concreteness) to perform well on semantics tasks and the methods ranked inputs by syntactic difficulty (syntactic complexity and surprisal) to perform better on syntax tasks.

Final results from all four CL methodologies as well as the two baselines are displayed in Table 1. Each of the four methods generally equaled or outperformed the baselines, although the Random CL baseline performed significantly worse than all the other methods on BLiMP and No CL did significantly worse on BLiMP Supplement. All models except the Random CL baseline performed roughly the same on the BLiMP dataset, whereas their scores significantly differed on the BLiMP Supplement and EWoK datasets. There are a few tasks in BLiMP such as "wh_questions_subject_gap_long_distance_filtered" that are extremely difficult and skew the averages for all models downwards. As we hypothesized, we found that the models that focused on syntax performed better on BLiMP and the semantics-based models performed better on EWoK and BLiMP Supplement.

MMM, being a more robust system that takes both semantics and syntax (through POS tagging) into account, performed the best on average. We note that unlike the other three CL methods, this method saw all of the data for each epoch, even though the masking probabilities were different. It is likely that this additional information that the model received during training improved its performance. However, this didn't appear to help with the BLiMP tasks. Looking at the subtasks in BLiMP,

	EWoK (↑)	BLiMP (↑)	BLiMP Supplement (↑)
No CL	0.4993	0.4962	0.4266
Random CL	0.4979	0.4295	0.4967
Surprisal	0.4976	**0.5150**	0.4612
Concreteness	**0.6536**	0.5025	0.5842
Syntatic Complexity	0.4933	0.5101	0.4675
MMM	0.6459	0.5071	**0.6063**

Table 1: Curriculum Learning results for baselines and 4 CL methodologies. Baselines include a BabyBERTa model trained from initialization for 9 epochs with no CL, and a BabyBERTa model trained from initialization trained on 3 random curricula with 3 epochs per curriculum.

we find huge variance in the results from 0.96 down to 0.04. We attribute this behavior to the nature of the tasks and how they align with the training strategy. The best performing task was for "reflexive binding" in relatively short sentences where simple, local agreement cue ("him-/her-") is enough. Early MMM stages focused on high-frequency content nouns, pronouns. Tokens such as "himself / herself / the / a" are therefore both seen often and practiced in isolation before harder material appears. On the other hand, the worst performance for this model was on the "wh-vs-that" task that required tracking with a long-distance filler-gap and "NPI licensing" with negation. We think this is mainly because crucial tokens for these tasks exist in different POS/semantic buckets introduced in different stages. MMM trains them to baseline competence separately, but the structure of the curriculum does not allow the model to focus on their interaction.

The concreteness model was another high performing model on the EWoK and BLiMP Supplement datasets due to its semantic focus. However, it had low performance on the BLiMP tasks, particularly having trouble with tasks such as local agreement. This model was the only one (other than the random CL baseline) that did not expressly take syntax into account at all. It is possible that more concrete words appear in syntactically "easier" sentences, but concreteness and syntactic simplicity are not obviously related. Therefore, it is likely that the model saw both syntactically easy and hard inputs throughout all of the curricula, and was able to pick up on semantically tangible objects first due to the ordering of data. It is also likely that the amount of OOV words hindered this model and that a more thorough concreteness ranking method would lead to even higher performance. However, even with the limits of the concreteness rating dataset, this method of creating curricula is clearly well aligned with the EWoK task–which focuses on concepts such as concreteness.

The surprisal model was the highest performer on the BLiMP dataset, although not significantly so. Surprisal reflects token probability in context and therefore theoretically captures both semantics and syntax. However, we hypothesize that because the trigram probabilities were calculated from a relatively small dataset, it likely did not capture much semantics and rather reflects syntax difficulty via input length. It did well on BLiMP tasks that involved agreement and coordination, demonstrating that it learned simple rules about English syntax, but performed poorly tasks that involved large syntactic gaps. Those inputs would have had a high surprisal due to their length and would have been shown to the model only in the last curriculum. The syntactic complexity model also slightly improved upon the baseline for BLIMP and was outperformed on EWoK and BLiMP Supplement. The syntactic complexity rankings are highly dependent on model parameters, and we hypothesize that hyperparameter optimization of the scoring model might have slightly improved the scores. However, since the surprisal-based methodology only slightly better, we are not convinced that any gains from a hyperparameter search would be as robust as switching ranking methodologies entirely.

In order to analyze the overall utility of the stages of CL, we explore the performance of the models on the test datasets after each stage of training, as shown in Figure 3. From this we can see that increasing the difficulty of the training examples rarely helps for BLiMP, was consistently helpful for BLiMP Supplement, and was only helpful for EWoK with the methodologies that were focused on semantics. This indicates that utility of CL is correlated with the final task of the model. Figure 3 combined with the No CL baseline performance on

the BLiMP dataset shows that while CL might yield a performance boost for syntax tasks, that boost is minimal especially in comparison to semantics tasks.

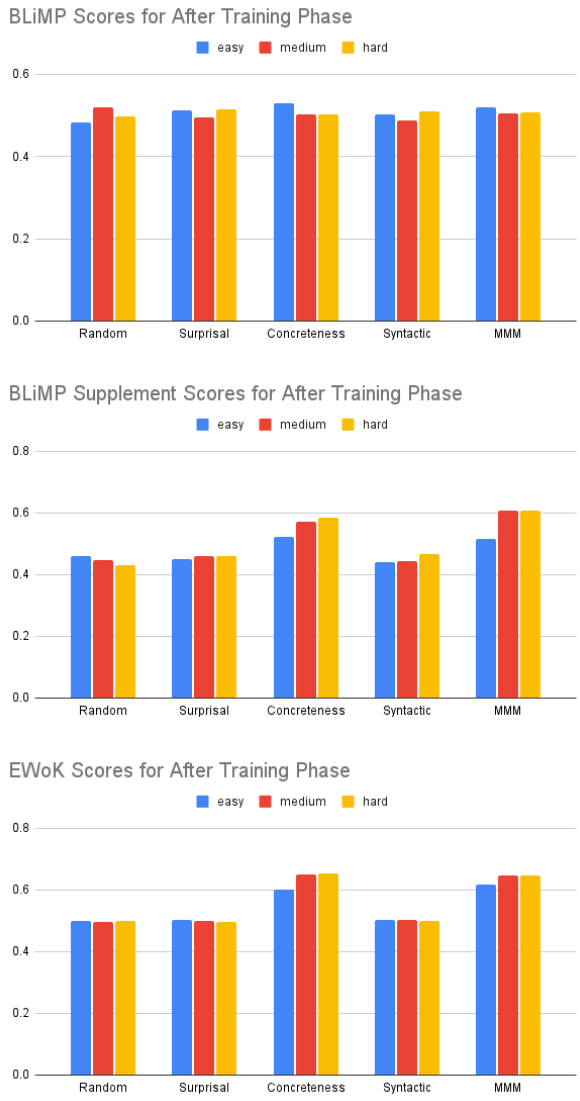

Figure 3: Model performance on evaluation tasks after each training phase. Increasing bars from left to right indicate that CL helped the model learn the task.

5 Discussion

We can clearly divide our methodologies into "semantics-" and "syntax-focused" bins. The models performed the best on the evaluation tasks that their training methodologies were most aligned with. However, these results clearly show that semantics-focused methodologies have a much larger effect on their aligned tasks than their syntax-focused counterparts, with the EWoK and BLiMP Supplement scores showing significant improvements over the baselines. For all of the datasets, a random performance is 50%, as all of these tasks require selecting a "better" sentence between the two. This means that the random CL baseline performed worse than random on BLiMP and the no CL baseline performed worse than random on BLiMP Supplement. For BLiMP, even our highest performing model (surprisal) barely performed higher than random. The final results shown in Table 1 are averages across the tasks for the dataset. All of the models, for some BLiMP task or another, performed significantly worse than random. For a model to perform significantly worse than 50% suggests that some tasks within these datasets are particularly tricky–meaning that perhaps the more "awkward" constructions are the ones that are actually grammatically or conceptually correct. These tasks drag down the averages, as they measure "hard" aspects of English grammar. The syntax-focused methodologies performing worse than random on BLiMP Supplement suggests that they are *losing* a sense of semantic understanding for those tasks in order to gain performance on the simpler BLiMP tasks. On the other hand, having the models focus on semantics did not cause them to significantly forget syntactic concepts. Although syntax and semantics are conceptually different, they are clearly interrelated. These results imply that in order to perform well on these tasks, ordering the curricula by semantics appears to capture the information that is needed for both the BLiMP and EWoK tasks.

Finally, we note that since the baselines both perform significantly worse than average on some of the datasets, it could be that the MLM training approach is not the optimal method to teach a model about the framework of language. The recent success of causal language models suggests that the task of predicting the next word rather than predicting randomly masked words might be more aligned with the overall concept of language modeling.

These experiments show that CL can be an effective method to help a model converge faster and generalize better under extreme resource limitations such as in the BabyLM challenge. CL methodologies that use semantics seem to be a particularly helpful approach for semantics-based tasks. The difference in final performance across evaluation datasets demonstrates that different CL methodologies can significantly improve model performance *depending on the difficulty of the final task and the similarity between the methodology and the final*

task. Therefore, in order to optimize the performance of a model trained using CL, one should align the method of curriculum creation with the intended task of the final model.

6 Limitations

The models in this experiment were trained for a total of nine epochs with no hyperparameter optimization. It can be expected that more training time and hyperparameter tuning would improve the performance of all models. The curriculum splitting methodologies all have individual limitations as well. Firstly, the Random CL baseline should be run multiple times because the current curriculum split seems to favor semantics over syntax.

Surprisal can be calculated using a neural language model, leading to more robust probabilities. Beyond the probabilities, alternative methods for calculating surprisal for a series could improve performance: averaging the token surprisal would allow for an understanding of information content despite length, while more complex methods such as using sequential unmasking (Kauf and Ivanova, 2023) might lead to a more accurate measurement of information content in a series.

For the concreteness method, OOV words could be assigned a score rather than just skipping all of them, which could possibly result in better performance. Additionally, taking N-grams into account is a possible way to improve the model, where our original method was mainly focused on the concreteness scores for individual word.

The syntactic complexity model is highly dependent on the parameters chosen, particularly the tree depth and clause count. Increasing the tree depth weights could allow the model to better capture hierarchal structure and lowering the clause counts weights could make the model less sensitive to complex subordination, which could better balance the models performance across the evaluation sets. SpaCy can be substituted for a more accurate parser like the Stanford Parser, which is computationally costlier but can better handle complex and/or ambiguous sentence structures. Incorporating additional complexity features such as average dependency distance or branching factor could provide a more nuanced view of complexity. Furthermore, implementing a proper assessment of the confidence of each parse would help identify problematic structures that might skew the scores.

For MMM, the current approach assigns fixed probabilities to all tokens (either 0.4 or 0.15). The model can be further improved by assigning different probabilities based on the actual frequency of the tags in the corpora. The frequencies can also be a better measure of which tokens to include in which stage instead of a subjective assessment of how easy tokens are. In addition, further finetuning with uniform probabilities at the end can enforce more focus on cross bucket tokens interactions.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 41–48, Montreal Quebec Canada. ACM.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904–911.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. (arXiv:2502.10645). ArXiv:2502.10645.

Mohamed Elgaar and Hadi Amiri. 2023. Ling-cl: Understanding nlp models through linguistic curricula. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 13526–13542, Singapore. Association for Computational Linguistics.

Henok Ghebrechristos and Gita Alaghband. 2019. Optimizing training using information theory-based curriculum learning factory. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, page 1525–1530, Portland, OR, USA. IEEE.

Edward Gibson, Richard Futrell, Steven P. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23(5):389–407.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001 - NAACL '01*, page 1–8, Pittsburgh, Pennsylvania. Association for Computational Linguistics.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, page 231–240, Singapore. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. Babyberta: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, page 624–646, Online. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. (arXiv:2405.09605). ArXiv:2405.09605.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, page 925–935, Toronto, Canada. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

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:1907.11692). ArXiv:1907.11692.

Allan Paivio, John C. Yuille, and Stephen A. Madigan. 1968. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of Experimental Psychology*, 76(1, Pt.2):1–25.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 2699–2712, Online. Association for Computational Linguistics.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pretraining cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, page 174–188, Miami, FL, USA. Association for Computational Linguistics.

Ryan Spring and Matthew Johnson. 2022. The possibility of improving automated calculation of measures of lexical richness for efl writing: A comparison of the lca, nltk and spacy tools. *System*, 106:102770.

Benedikt Szmrecsanyi. 2004. On operationalizing syntactic complexity. In *Le poids des mots. Proceedings of the 7th international conference on textual data statistical analysis. Louvain-la-Neuve*, volume 2, pages 1032–1039.

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

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*, page 6095–6104, Online. Association for Computational Linguistics.