Progressive Learning in LLMs Using Structured Grammar Books

Rishabh Patil
New York University
rbp5812

Mustafa Poonawala
New York University
msp9471

Airu Liu
New York University
al7038

Devyani Hebbar
New York University
dh3677

1 Introduction

The development of Large Language Models (LLMs) has significantly advanced natural language processing, enabling machines to perform tasks such as translation, summarization, and question-answering with remarkable proficiency. Traditional LLM training approaches often rely on vast, unstructured datasets, necessitating substantial computational resources and time. However, recent research suggests that a more structured and efficient learning paradigm—one that mirrors human language acquisition—can enhance model performance while reducing training complexity. In this paper, we propose a progressive training framework for LLMs that **emulates human language acquisition** through the pre-training of a sequence of grammar books and incorporating morphological word embeddings. This approach aims to improve model competence and efficiency by **optimizing training methodologies through linguistic tagging and curriculum-based training**.

2 Related Work

2.1 Less is More

In machine learning, the Minimum Description Length (MDL) principle posits that the optimal model for a given dataset is the one that offers the most concise yet accurate representation, effectively balancing complexity and generalization (Rissanen, 1978). The "Less Is More" principle emphasizes efficiency over scale, where **selective data usage, task-specific training, and optimized architectures** lead to superior performance with fewer computational resources.

Data selection enhances model generalization by prioritizing high-quality, diverse, and informative data over vast redundant datasets (Sorscher et al., 2022), ensuring that models learn from critical examples and reducing overfitting risks. Studies have shown that **fine-tuned smaller models**, such as BioBERT for biomedical NLP (Lee et al., 2020) and legal-specific LLMs, often **outperform larger general-purpose models in specialized domains**. Additionally, **efficient architectures**—via techniques such as **model compression, pruning, and parameter sharing**—reduce model size while preserving knowledge, as demonstrated by DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2020), and DeepSeek. By integrating data selection, targeted training, and efficient architectures, machine learning can achieve state-of-the-art results with lower computational costs, challenging the assumption that bigger models are inherently better.

2.2 Curriculum Learning

Traditional LLMs often process text by treating each sequence independently, without considering the overarching structure inherent in human language. In contrast, human language acquisition involves a **gradual and structured learning process**, where individuals learn through social interactions, beginning with simple sounds and progressing to complex sentences. This developmental approach enables a deep understanding of context and meaning, which purely statistical models may lack.

Building upon foundational work that demonstrated the benefits of starting with simplified inputs (Elman, 1993; Hale, 2001; Reali and Christiansen, 2005), Bengio et al. (2009) introduced the term *curriculum learning* to formalize this training strategy. Curriculum learning involves presenting training data in a meaningful order—**starting with easier aspects of a task and progressively introducing more complex elements**. Subsequent research has expanded on this approach, with Narvekar et al. (2020) providing a comprehensive framework in reinforcement learning, Wang et al. (2020) offering an extensive review across various machine learning fields, and Warstadt (2023)

demonstrating sample-efficient pretraining using developmentally plausible corpora. These studies collectively underscore the efficacy of **curriculum learning in improving both the speed and quality** of model training.

3 Methodology

3.1 Dataset Preparation

The foundational dataset for this project was constructed from the *New Concept English* textbook series (Volumes 1–4). Each team member was responsible for digitizing and preprocessing the content of one volume using OCR (Tesseract) and structured normalization steps:

- Correction of OCR errors (e.g., ligature fixes, punctuation standardization).
- Consistent formatting of dialogue, currency, and number formats according to British English conventions.
- Structuring lessons with metadata tags for grammar concepts and composition exercises.

This resulted in a machine-readable, curriculum-organized dataset spanning 345 lessons (less than 20k sentences), intended for progressive training from basic to advanced language constructs. Additionally, the TREC Question Classification Dataset was selected as the primary external benchmark to evaluate model performance on text classification tasks.

3.2 Model Selection: GPT-2 and T5-base

For this study, we selected GPT-2 and T5-base as the primary backbone models. GPT-2 is a decoder-only Transformer architecture trained with a left-to-right autoregressive objective, where each token is generated conditioned only on previous tokens. In contrast, T5-base follows an encoder-decoder Transformer design, where the encoder processes the full input sequence and the decoder generates output tokens conditioned on both the encoded input and previously generated outputs.

The choice of GPT-2 and T5-base enables a comparative investigation across two distinct modeling paradigms: generation-only models and sequence-to-sequence models. Decoder-only architectures, like GPT-2, are lightweight and efficient for causal language modeling and classification through prompt-based generation. Encoder-decoder architectures, like T5, provide stronger bidirectional context modeling and are naturally suited for supervised tasks such as translation, summarization, and classification. An overview

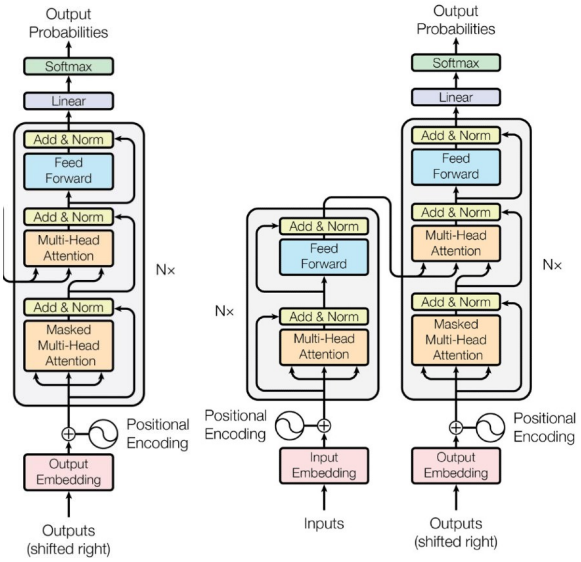

Figure 1: Comparison of Transformer architectures: Left — GPT (decoder-only); Right — T5 (encoder-decoder).

of the architectural differences between encoder-decoder and decoder-only models is shown in Figure 1. Both GPT-2 and T5-base have moderate parameter sizes (GPT-2 ∼117M and T5-base ∼220M), balancing computational feasibility with sufficient model capacity for progressive grammar-based training.

Model configuration details are summarized in Table 1.

Model	Params (M)	Hidden Size	Layers	Heads
GPT-2	∼117	768	12	12
T5-base	∼220	768	12	12

Table 1: Model configuration details for GPT-2 and T5-base.

3.3 Baseline Model Training

To establish baseline performance, we directly loaded the pretrained GPT-2 and T5-base models from the Hugging Face Transformers library and fine-tuned them on the TREC classification dataset. No modifications were made to their architecture or objective during this stage. The models were loaded as follows:

```
# Load T5-base
t5_model = AutoModelForSeq2SeqLM.from_pretrained("t5-base")
# Load GPT-2
gpt2_model = AutoModelForCausalLM.from_pretrained("gpt2")
```

These baseline models serve as reference points for evaluating the effectiveness of our syntax-enhanced variants. Their corresponding performance results are presented in Section 4.4. .

3.4 Syntax-Based Model Development

After establishing baselines, we introduced syntax-enhanced models to investigate the impact of explicit grammatical information on model performance.

3.4.1 Syntactic Feature Extraction

In the syntax-augmented approach, each input token was paired with additional syntactic features: Specifically, for each token, we extracted:

- Part-of-Speech (POS) tags,
- Dependency relations (DEP),
- Named Entity Recognition (NER) tags, and
- Morphological features (Morph, e.g., tense, number, case).

These features were fused into a single identifier per token using a 4-tuple `(POS, DEP, NER, Morph)`. The uniqueness of each syntactic embedding is defined by the specific combination of these four fields. For example, several combinations and their corresponding frequencies are:

```
('NOUN','root','O','Number=Sing')                    → 1529
('NUM','dep','CARDINAL','NumForm=Digit|NumType=Card') → 157
('PUNCT','punct','O','')                             → 30403
('VERB','parataxis','O','Mood=Imp|VerbForm=Fin')     → 47
```

Across all four volumes of the New Concept English textbook, we identified a total of 1,704 unique syntactic combinations. This reflects a rich distribution of grammar structures and lexical diversity. The frequency imbalance among these combos suggests that while some grammatical constructs are ubiquitous (e.g., punctuation or root-level verbs), others are semantically or morphologically rare. This compositional variety allows the model to learn fine-grained distinctions between syntactic roles and grammatical nuances during training.

3.4.2 Syntax-Aware Model Construction

This syntax-augmentation strategy was applied separately to two base architectures:

- **SyntaxGPT**: a modified version of GPT-2, and
- **SyntaxT5**: a modified version of T5-base.

For both SyntaxGPT and SyntaxT5, all pretrained weights were removed, and the models were initialized randomly from scratch using their original configuration files. No prior knowledge from external corpora was preserved. The models were instantiated as follows:

```
# Randomly initialized GPT-2
gpt2_config = GPT2Config.from_pretrained("gpt2")
gpt2_model = GPT2LMHeadModel(gpt2_config)
# Randomly initialized T5
t5_config = T5Config.from_pretrained("t5-base")
t5_model = T5ForConditionalGeneration(t5_config)
```

We integrated the syntax combination features (POS, DEP, NER, Morph) described previously into the token input space by augmenting the embedding layers. The following table summarizes the structural modifications:

Model	Embedding Structure
GPT-2	Token + Position $\rightarrow$ Projected [Token $\|$ Syntax] + Position
T5-base	Token only $\rightarrow$ Projected [Token $\|$ Syntax]

Table 2: Embedding structure before and after syntax augmentation. "$\|$" denotes concatenation before linear projection.

3.4.3 Training Strategy

Both SyntaxGPT and SyntaxT5 underwent a two-phase training process:

1. **Curriculum Pretraining**: Progressive training on the structured *New Concept English* lessons, advancing from simple to complex language structures.
2. **Fine-Tuning**: Subsequent fine-tuning on the TREC dataset specifically for question classification tasks.

In addition to full curriculum pretraining, we also conducted experiments where SyntaxGPT and SyntaxT5 models were fine-tuned directly on the TREC dataset without any prior curriculum pretraining. This allowed us to isolate and evaluate the direct impact of syntactic features independently of staged linguistic exposure.

In addition, the choice of architectures also allowed us to explore the role of syntactic features across different modeling paradigms. SyntaxGPT has a decoder-only transformer architecture, and generates outputs autoregressively. In contrast, SyntaxT5 has an encoder-decoder transformer architecture. Comparing the two offers insight into how syntax influences models with fundamentally different information flow patterns.

3.5 Evaluation Metrics

Performance across all experiments was assessed using standard classification metrics:

- Accuracy,
- Recall,
- F1 Score,
- Pre-training Time,
- Trec Classification Time.

Comparative evaluations between baseline and syntax-augmented models allowed us to measure the benefits of incorporating explicit grammatical information into transformer-based architectures.

4 Results and Discussion

4.1 GPT-2

The SyntaxGPT model, initialized with random weights and pretrained on a structured grammar-based curriculum, demonstrated the ability to acquire basic syntactic patterns. However, when evaluated on the downstream TREC classification task, it achieved an accuracy of 63%, which was lower than the 71% accuracy obtained by a randomly initialized GPT-2 model trained directly on TREC without grammar pretraining.

This result suggests that while the grammar curriculum provided useful syntactic scaffolding, it may have introduced a bias toward narrow linguistic regularities, limiting adaptability to open-domain classification tasks. Additionally, evaluation loss metrics collected during training further support this interpretation. In many curriculum lessons, the evaluation loss remained unusually high—many in the range of 5 to 7—even after extended training.

For classification tasks such as TREC, well-trained models commonly achieve evaluation losses closer to 1.5 to 2.5. The elevated losses suggest that the model struggled to produce confident and accurate predictions, possibly due to ineffective transfer from the grammar pretraining phase or insufficient fine-tuning.

It is also possible that the fine-tuning stage itself was not sufficiently optimized. Factors such as limited training duration, early stopping, or suboptimal learning rates may have prevented the model from adapting effectively to the classification objective.

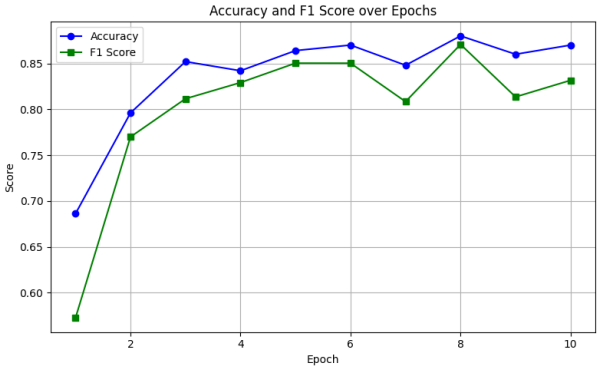

Figure 2: SyntaxT5 model accuracy (left) and F1 score (right) progression over training epochs.

Taken together, these findings highlight several important limitations. While syntax-guided curriculum training offers a principled approach to pretraining, it alone may not provide enough semantic breadth or task alignment to ensure strong downstream generalization. Future work should explore integrating more semantically rich and task-relevant data, improving fine-tuning strategies, and better aligning curriculum objectives with end-task goals. Despite the current performance gap, this experiment offers valuable insight into the challenges and potential of structured pretraining for low-resource language modeling.

4.2 T5

For our SyntaxT5 models, both accuracy and F1 scores improved substantially across training epochs, reflecting successful incremental learning facilitated by the curriculum structure. Specifically, accuracy increased from 68.6% at epoch 1 to 88% at epoch 8, as shown in Figure 2, demonstrating a clear progression in model competence. This progressive learning approach enhanced performance relative to initial baselines, although it did not surpass the final accuracy scores achieved by directly fine-tuned models, such as 97.6% for T5-base.

Despite this, the trajectory of improvement emphasizes the effectiveness of curriculum-based training, particularly in low-resource and sequential learning contexts. The validation loss steadily decreased from 0.826 to 0.6 over ten epochs, further indicating effective generalization and progressive mastery of the linguistic structures. Notably, evaluation loss for the SyntaxT5 model remained within a moderate range—mostly between 3 and 4—compared to the elevated 5 to 8 loss ob-

Model	Accuracy	Recall	F1-score	Pretraining Time	Classification Duration	
					Training	Inference
GPT-2 (Pre-trained)	0.954	0.940	0.922	30 d	387.89 s	32.87 s
T5 (Pre-trained)	0.976	0.8476	0.8387	2.5 d	236.37 s	1.96 s
GPT-2 RAW + GRM + EMBs	0.636	0.201	0.223	1.25 h	288 s	
T5 RAW + GRM + EMBs	0.870	0.846	0.850	3 h	114 s	
Optimal RAW(GPT-2, T5) + GRM	*Details in Section 4.3*					

Table 3: Design of the Evaluation Models

served in the SyntaxGPT setup, suggesting that SyntaxT5 benefited more consistently from curriculum pretraining.

Similarly, the F1 score increased consistently, reaching 0.87 at epoch 8 in Figure 2, highlighting strong improvements in both precision and recall compared to earlier training stages. Detailed analysis of the model's predictions revealed increasing alignment between predicted and true labels, particularly after epoch 6, suggesting enhanced model confidence and classification capability over time. This outcome supports our broader hypothesis that structured linguistic signals can compensate for limited data and computational resources when integrated effectively into the training process.

4.3 Syntax Embedding Ablation

To further isolate the contribution of syntactic features, we conducted ablation experiments (both pretrained for 10 epochs, parameters are proper selected but only for ablation study) comparing models trained with and without syntax embeddings on the TREC classification task. We compare their performance based on accuracy and F1 score to assess the impact of explicitly providing grammatical information to the model. This comparison allows us to evaluate how syntax influences convergence behavior, generalization, and model robustness.

Analysis The experimental results, illustrated in Figure 3, reveal a clear advantage for models incorporating syntax embeddings in the TREC classification task. Compared to the baseline model, the syntax-enhanced model not only converges more rapidly but also achieves higher final performance in both accuracy and F1 score.

The inclusion of syntax-aware information provides an important inductive bias that helps the model better capture the underlying structure of the input questions. Instead of relying solely on surface-level token patterns, the syntax-augmented model is able to exploit deeper linguistic cues — such as grammatical relations and entity types — which are critical in question understanding and categorization tasks.

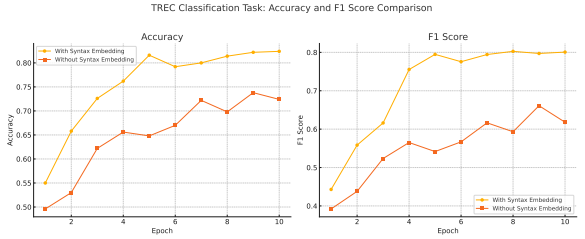

Figure 3: Comparison of Accuracy and F1 Score on the TREC classification task between models with and without syntax embedding.

Moreover, the training dynamics indicate that syntax embeddings contribute to greater stability during optimization. The model with syntax features shows smoother and more consistent improvement, suggesting that the added structural information acts as a form of regularization, reducing overfitting and making the learning process more robust.

Overall, these findings support the view that enriching token representations with syntactic features can significantly enhance the learning of classification models, especially on tasks where linguistic structure is strongly correlated with the target labels.

4.4 Performance Comparison on the TREC Classification Task

Table 3 compares the performance of baseline pretrained models (GPT-2, T5) and experimental models trained from scratch with grammar embeddings (GRM + EMBs). Several key observations emerge from the comparison.

Without heavy pretraining, T5 combined with grammar embeddings achieves strong performance, reaching 87% accuracy with a high recall and F1-score. In contrast, GPT-2 trained under the same conditions exhibits significantly lower accuracy, poor recall, and low F1-score, despite similar training conditions. This indicates that the T5 architecture is more capable of benefiting from grammar-enhanced inputs when trained with limited resources.

In terms of classification duration, the T5 RAW model also completes both training and inference more efficiently than the GPT-2 RAW model. The total time required for T5 is notably shorter, suggesting that it not only achieves higher performance but also converges faster under the grammar-augmented setting.

Overall, the results show that adding grammar embeddings significantly improves the learning capacity of raw models. However, the extent of improvement is highly architecture-dependent. T5 demonstrates a much stronger ability to leverage grammatical information for classification tasks compared to GPT-2 when trained without large-scale pretraining.

5 Limitations and Future Works

5.1 Limitations

Despite the promising results observed with syntax-augmented models, several limitations were identified during experimentation.

1. **Curriculum Complexity Saturation**
   The progressive curriculum method relies heavily on the sequencing of instructional material. While early lessons yield significant improvements, gains may plateau as curriculum complexity increases, suggesting diminishing returns from increasingly advanced grammar input without concurrent expansion in semantic diversity.

2. **Limited Domain and Objective Scope**
   Pretraining on structured grammar-focused material, such as span-masking over textbook sentences, restricts model exposure to real-world variability. These sentences are often short, well-formed, and semantically simple. As a result, models trained this way may struggle to generalize to the diverse phrasings, topics, and reasoning patterns found in open-domain tasks like TREC classification.

3. **Data Scale Constraints**
   The curriculum dataset, though pedagogically rich, comprises only a few thousand examples. In contrast, baseline models like GPT-2 are pretrained on hundreds of millions of tokens from large-scale datasets (e.g., WebText), enabling them to develop broad, transferable representations. This scale mismatch limits the ability of the syntax-based models to learn robust, transferable representations, particularly when trained from scratch.

4. **Model Architecture Sensitivity**
   T5, with its encoder-decoder structure, demonstrated greater resilience to limited pretraining, leveraging the explicit separation of input encoding and output generation. In contrast, GPT-2's decoder-only design, optimized for next-token prediction over massive corpora, failed to adapt efficiently when trained solely on small, structured datasets. This architectural sensitivity underscores the critical dependence of decoder-only models on scale and diverse input for effective learning.

5.2 Future Works

Future work will focus on several promising directions to extend the effectiveness of progressive curriculum training. First, we plan to expand the curriculum with additional layers of linguistic complexity, including more advanced syntactic structures, semantic nuances, and discourse-level coherence features. This will allow models to gradually acquire richer linguistic competence beyond basic grammatical patterns.

Second, we will incorporate dictionary-based pretraining by leveraging resources such as Wiktionary to provide morphological enrichment. By systematically introducing information about word forms, inflections, and derivations, we aim to enhance the model's understanding of fine-grained linguistic variations, which may be especially beneficial for tasks requiring morphological sensitivity.

Third, comprehensive ablation studies will be conducted to isolate and quantify the contribution of individual components, such as grammar embeddings, curriculum progression stages, and input augmentation strategies. This will provide deeper insight into which aspects of the progres-

Task	**Assignee**	
OCR Text Collection	All	
Morphology Tagging	All	
Model Prototypes in Python	All	
Models Training		
1. GPT-2	Mustafa	Devyani
2. T5	Rishabh	Airu
3. GPT-2 RAW + GRM + EMBs	Airu	Mustafa
4. T5 RAW + GRM + EMBs	Devyani	Rishabh
5. Max (GPT-2, T5) RAW + GRM	All	

Table 4: Task Division

sive methodology most significantly impact model performance and learning efficiency.

In addition, fine-tuning hyperparameters—including learning rate schedules, optimizer settings, and batch sizes—as well as adjusting the granularity of curriculum steps may further optimize both convergence speed and final accuracy.

Finally, a major future direction is applying the progressive curriculum framework to grammatical error correction (GEC) tasks. Given that GEC inherently requires a deep understanding of grammatical structures and the ability to recognize and correct subtle linguistic errors, it aligns naturally with the goals of progressive training. We anticipate that the structured acquisition of grammatical competence fostered by the curriculum approach will significantly enhance GEC performance, particularly in low-resource or domain-specific settings.

Overall, the progressive learning methodology presents a compelling alternative for structured language model training, emphasizing both computational efficiency and incremental linguistic skill acquisition.

6 Conclusion

This study investigated grammar-enhanced progressive training under constrained pretraining settings and conducted syntax ablation experiments to assess the contribution of syntactic features. The experimental results reveal a nuanced picture.

When trained from scratch, T5 models with grammar embeddings achieved moderate success, demonstrating relatively strong classification performance with limited pretraining. In contrast, GPT-2 models performed poorly even with grammar augmentation, indicating that decoder-only architectures are significantly more reliant on large-scale pretraining. These results highlight that grammar-enhanced progressive training is not universally effective across architectures and is currently better suited to encoder-decoder models like T5.

In addition to overall performance comparisons, the syntax ablation experiments provide important insights. Models augmented with syntax embeddings consistently outperformed their counterparts without syntax, achieving faster convergence, higher final accuracy, and improved F1 scores on the TREC classification task. The inclusion of syntactic information acts as an effective inductive bias, helping models better exploit the grammatical structure of input texts. Furthermore, syntax-aware models exhibited greater training stability, suggesting that structural information serves as a useful form of regularization.

Overall, while grammar-enhanced progressive learning shows promise in specific contexts, particularly when combined with architectures capable of leveraging syntactic information, it is not a replacement for pretraining at scale. Future work will focus on expanding the linguistic complexity of the curriculum, enriching morphological knowledge through dictionary-based resources, conducting more extensive ablations, and applying the progressive training framework to grammatical error correction (GEC) tasks, where incremental grammatical competence acquisition is especially crucial.

7 Contribution and Timeline

7.1 Task Distribution

Task division will be organized as shown in Table 4. Each task will be assigned based on team expertise to ensure efficiency and quality in model training and evaluation.

7.2 Progress Timeline

- **Week 1**: Conducted project planning and literature review on curriculum learning and linguistic embedding strategies.
- **Week 2**: Performed OCR-based digitization of New Concept English textbooks (Volumes 1–4) using Tesseract. Completed preprocessing and normalization of digitized texts, including error correction and metadata tagging. Structured the dataset and implemented a metadata schema for lesson organization and traceability.
- **Week 3**: Finalized benchmark dataset selections (TREC) and selected baseline models (GPT-2 and T5-base) for the experimental phase by fine tuning baseline models (GPT-2, GPT-Neo, GPT-2 Medium, T5-small, T5-base) on the TREC datasets. Conducted initial evaluation and error analysis of baseline performance using classification and correction metrics.
- **Week 4**: Progressive Training Initiation: Sequentially trained experimental models using the structured grammar curriculum, starting from elementary to advanced volumes.
- **Week 5**: Model Evaluation and Comparison: Evaluated experimental models on TREC using standard NLP metrics (Accuracy, F1, and Recall), and benchmarked them against baseline models.
- **Week 6**: Ablation Testing: Conducted ablation experiments to assess the isolated impact of curriculum structure and morphological embeddings on performance.
- **Week 7**: Error Analysis and Refinement: Analyzed failure cases to inform prompt tuning, embedding adjustment, and model fine-tuning strategies.
- **Final Report and Presentation Preparation**: Consolidated findings, visualizations, and insights into the final project report and prepared the presentation.

References

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

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

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of NAACL*, pages 1–8.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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

Florencia Reali and Morten H. Christiansen. 2005. Uncovering the richness of the stimulus: Structure dependence and indirect statistical evidence. *Cognitive Science*, 29(6):1007–1028.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter.

Ben Sorscher, Mario Geiger, Jakob H. Macke, Surya Ganguli, and Ari S. Morcos. 2022. Beyond neural scaling laws: Beating power law scaling via data pruning.

Xin Wang, Zhaoxuan Huang, and Ji Liu. 2020. A survey on curriculum learning in machine learning.

Alex et al. Warstadt. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of EMNLP*.