Text Complexity Matters Less Than Information Content When Pretraining Language Models

Anonymous ACL submission

Abstract

Improving the quality and size of the training corpus is known to enhance overall downstream performance of language models on general language understanding tasks. However, the impact of text complexity on downstream performance has been less studied. Text complexity refers to how much easier or harder a text is to read compared to others, taking into account lexical (e.g., vocabulary choice), syntactic (e.g., sentence structure), and semantic complexity (e.g., information content), among others. In this work, we focus on reducing lexical and syntactic complexity, while controlling for semantic complexity. We ask two core questions: (1) Does text complexity matter in pretraining? and (2) How does the text complexity of our pretraining corpora affect the performance of language models on general language understanding tasks? To answer these questions, we simplify human-written texts using a large language model (with the goal of retaining the information content) and pretrain GPT2-small models on both the original and simplified versions. We show empirical evidence that lexical and syntactic complexity do not significantly affect performance on general language understanding tasks, emphasizing the importance of information content when pretraining language models.

1 Introduction

Let's compare two versions of text:

(A) As the sunset cast its warm orange glow over Manila Bay, people relaxed on the sideline benches, enjoying the peaceful view of the sunset.

(B) The sunset gave Manila Bay a warm, orange light. People sat on the benches and enjoyed the view of the sunset.

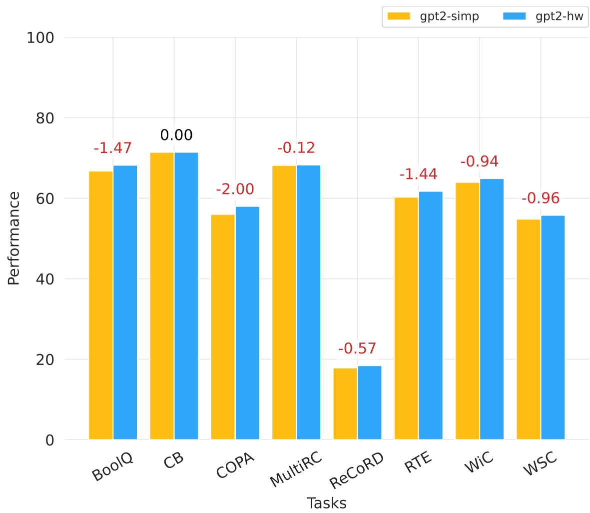

Figure 1: Relative performance of `gpt2-simp` (trained on simplified texts) vs. `gpt2-hw` (trained on human-written texts) across the 8 SuperGLUE tasks shows minimal differences, suggesting text complexity has little impact on general language understanding. Accuracy is used for all tasks.

The two versions convey the same core meaning, but one uses more nuanced, complex language, whereas the other is simpler and less nuanced. This can be likened to lossy compression, where version (B) requires fewer bits to represent the information in (A) but loses some of its nuance. It compresses by using common words and simpler sentence structures while retaining the core information.

What if our corpus is more like (B)? Can we still learn useful representations by training solely on simplified text with a simpler vocabulary and sentence structure? To answer this, we explore the relationship between text complexity and downstream performance, focusing on lexical and syntactic complexity while keeping information content mostly constant.

It is well-known that language models acquire world knowledge during pretraining (Petroni et al., 2019; Roberts et al., 2020; Zhang et al., 2021; Wei et al., 2022), and transfer learning is more effective when the pretraining corpus aligns with the target task domain (Ruder and Plank, 2017; Gururangan et al., 2020). For example, pretraining on medical texts and fine-tuning on medical tasks is more effective than pretraining on social media texts. In other words, a model's knowledge significantly impacts its downstream performance. Therefore, to isolate the effect of text complexity, it's crucial to control for information content. In this paper, we ask two core questions:

(1) Can we learn useful representations in our base models by training solely on simpler text, with simpler vocabulary and sentence structure?

(2) How does the text complexity of our pretraining corpora impact language model performance on general understanding tasks?

To answer these questions, we collect human-written texts and transform them into simpler language using a Large Language Model (LLM) while preserving the core information content. We pretrain GPT2-small models (Radford et al., 2019) from scratch in two controlled setups, one on human-written (more complex) texts and another on the simplified version of the same texts. Lastly, we finetune and evaluate these models on the SuperGLUE benchmark (Wang et al., 2019), which is a collection of general language understanding tasks.

Our empirical evidence shows that reducing lexical and syntactic complexity doesn't significantly impact performance on general language understanding tasks. This highlights that, at the pretraining stage, the content of the training data matters more than its form.

2 Related Work

Text complexity (also known as readability). Text complexity or readability refers to how difficult a text is to understand (DuBay, 2004), influenced by linguistic factors such as word choice (e.g., "utilize" vs. "use"), sentence structure (complex vs. simple), and content type (academic vs. children's books) (Dale and Chall, 1948, 1949; Graesser et al., 2004). Although other factors such as the reader's background knowledge also affect readability (Ozuru et al., 2009), this work focuses solely on linguistic aspects.

Several metrics have been proposed for readability such as Flesch Reading Ease (Flesch, 1948) (FRE), Dale–Chall (Dale and Chall, 1948), and SMOG (Mc Laughlin, 1969). These formulas rely on surface-level features like text length, word count, and word length. While they're useful estimates, they don't tell the whole story. This limitation has prompted the use of machine learning and deep learning approaches (Hancke et al., 2012; Imperial and Ong, 2021; Chatzipanagiotidis et al., 2021; Imperial, 2021; Meng et al., 2020) to capture features beyond the surface level, such as coherence and writing style. More recently, researchers have begun exploring the use of Large Language Models (LLMs) for estimating readability (Trott and Rivière, 2024; Lee and Lee, 2023; Rooein et al., 2024). LLMs have shown strong correlations with human judgments compared to traditional formulas even without explicit finetuning (Trott and Rivière, 2024). However, using an LLM to score a large corpus is costly. For this reason, we use FRE to measure the complexity of our corpus.

Text simplification. Text simplification (TS) aims to make text easier to understand while preserving content (Agrawal and Carpuat, 2023; Alva-Manchego et al., 2019; Truică et al., 2023). While simplified texts tend to be shorter, that is not always the case (Shardlow, 2014). This is different from Text Summarization, where the goal is to shorten the text even if it changes the organization and content. Saggion and Hirst (2017); Shardlow (2014); Kriz et al. (2018) approached TS via word-substitution by replacing difficult words with easier synonyms using a lexicon. Other works approached TS as a translation problem using statistical machine translation (SMT) (Wubben et al., 2012; Scarton et al., 2018; Specia, 2010; Xu et al., 2016). Beyond SMT approaches, other works employed deep learning approaches such as encoder-decoder models (Zhang and Lapata, 2017; Alva-Manchego et al., 2019; Agrawal and Carpuat, 2023). Recent works explore LLMs for text simplification (Trott and Rivière, 2024; Imperial and Tayyar Madabushi, 2023; Farajidizaji et al., 2024; Padovani et al., 2024). While some works are concerned with simplifying texts to a specific grade-level, we are only concerned with making complex texts simpler, similar to Trott and Rivière (2024), which observes encouraging results on text simplification just by prompting LLMs. In this work, we use an LLM for text simplification.

Pretraining language models on simple texts. In recent years, there has been an increased interest in pretraining language models on simple texts. Zhao et al. (2023) found that a small language model (SLM), called BabyBERTa, trained on child-directed speech, performs on par with larger models on a set of probing tasks. Eldan and Li (2023) has shown that SLMs can learn to generate coherent and fluent text by training on synthetic texts of short stories that contain only words that 3- to 4-year-olds usually understand. Deshpande et al. (2023); Muckatira et al. (2024) has shown that SLMs pretrained on simplified language can achieve comparable performance to larger models when the problem is transformed to simple language. There is also a research community effort called "The BabyLM Challenge" (Warstadt et al., 2023; Hu et al., 2024) that emphasizes training on a fixed budget of 100 million words or less, sourced from texts intended for children, which are conceptually simpler.

Pretraining dataset design. Pretraining on massive texts is one of the main drivers of performance for modern language models (Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022). Pretraining data design choices such as domain composition, quality and toxicity filters, and collection date affect model performance in ways that cannot be adjusted by finetuning (Longpre et al., 2024). Most related to our work is Agrawal and Singh (2023) which studies the impact of corpus complexity on the downstream performance of language models. They observed that models trained on more complex texts (e.g., wiki), as measured by Flesch Reading Ease, yield stronger performance over less complex texts (e.g., children's books). While we are trying to answer the same question, the main difference between Agrawal and Singh (2023) and our work is that we preserve the information content and only vary the lexical and syntactic complexity.

Prior works have shown encouraging results for pretraining on simple texts. However, there is no work that looks at the direct impact of text complexity, more specifically at the lexical and syntactic level, on the downstream performance of language models at a relatively larger data scale i.e. 2.1B tokens and 5 domains. This calls for controlled experiments that will give evidence that a useful model can be learned by just training on simple texts.

3 Creating the Pretraining Datasets

3.1 Human-Written Corpora

We curated human-written English texts from two publicly available datasets: Dolma v1.6 (Soldaini et al., 2024) and Wiki-40B (Guo et al., 2020). Both have permissive licenses[1], and our usage complies with their intended purposes. The final corpus has around 2.34B tokens[2] uniformly distributed across 5 domains: web, books, social media, academic, and wiki. All domains are sourced from Dolma, except for wiki which is from Wiki-40B. We limit our dataset to 2.34B tokens because processing the full corpus would be too expensive. This number is based on Chinchilla Compute-Optimal guideline of 1:20 parameter-tokens ratio (Hoffmann et al., 2022) as a rough guideline[3]. According to this, if we're using GPT2-small with 124M parameters, 2.48B is a good dataset size.

Since Dolma and Wiki-40B are too large, we only process a subset of shards. For Dolma, initial subset per domain was picked manually (see Appendix A for more details). For Wiki-40B, we only use English subset. For each domain subset, we count the tokens and sample the longest documents within the 75th-100th percentile for Wiki-40B and the 50th-75th percentile for Dolma, continuing until we reach 468M tokens per domain. We sample within a specific percentile because outliers tend to occur on extreme ends. The sampling strategy prioritizes longer documents to enhance the models' exposure to extended texts, aiming to improve its ability to capture long-distance relationships between dispersed pieces of information.

3.2 Text Simplification via Large Language Model

We prompt Llama 3.1 8B instruction model (Grattafiori et al., 2024) to transform human-written texts into simplified texts. For efficient inference, we use the INT8 quantized version[4] of the model and vLLM (Kwon et al., 2023) as our LLM serving system. We discuss more about the prompt engineering and include the final prompt in Appendix B.

[1] ODC-BY license for Dolma, and Creative Commons for Wikipedia.

[2] We used GPT2 Tokenizer: https://huggingface.co/openai-community/gpt2.

[3] We initially used 117M as parameter count instead of 124M which is why our corpus is 2.34B.

[4] https://huggingface.co/neuralmagic/Meta-Llama-3.1-8B-Instruct-quantized.w8a8

We split the documents from the human-written corpora into paragraphs, resulting in a total of 28.5M paragraphs. We apply the transformation **paragraph-wise** because the model tends to summarize rather than simplify multi-paragraph documents. This approach preserves the original content and structure. However, not all paragraphs are transformed. This can happen under three conditions: (1) when a paragraph is too short relative to its full document; (2) when a paragraph is too long; or (3) when the transformation is significantly shorter or longer than the original text. In the case of (3), we revert to the original text in the final corpus. We include a more detailed breakdown of these conditions in Appendix C.

3.3 Resulting Simplified Texts

The final simplified corpus has around 2.12B tokens. There is a total of 28.5M paragraphs, of which 34.9% are not transformed (i.e., 22.21% are skipped and 12.69% are transformed but reverted back to the original). The domain distribution of the paragraphs that are not transformed are as follows: web (26.85%), books (25.49%), social media (21.90%), academic (6.97%), and wiki (18.80%). Overall, this accounts for 36.69% of total tokens of the final simplified corpus. Note that most of these texts are very short or very long inputs that are not informative (e.g., author names, table of contents, etc.), or already concise enough to require no further simplification.

To get a rough idea of what the simplified texts look like, see the following example:

> **Original**: Your comment really helped me feel better the most. I was sitting in my office, feeling so bad that I didn't say how inappropriate and out of line his comments were, and this helped.
>
> **Simplified**: Your comment really helped me feel better. I was feeling bad because I didn't speak up when someone made inappropriate comments.

4 Experimental Setup

In our study, we investigate the effect of text complexity on both the pretraining dynamics and downstream performance of language models. To do this, we compare models trained on human-written texts with those trained on simplified texts, conduct domain ablation experiments, and examine a curriculum approach that begins by presenting simplified texts to the model, followed by transitioning to complex texts.

4.1 Model Architecture and Training Details

We train GPT2-small models from scratch. Our configuration follows the standard GPT2-small setup: 124M parameter models with 12 transformer layers, 12 attention heads, and a hidden dimension of 768. These specifications are consistent with the original GPT2 publication (Radford et al., 2019) as implemented by HuggingFace[5]. All experiments are conducted using 8x P100 GPUs.

4.2 Pretraining Configurations

4.2.1 Human-Written vs. Simplified

We investigate how text complexity influences the model's ability to learn adaptable representations. Our primary motivation is to assess whether reducing lexical and syntactic complexity—while preserving semantic content—affects pretraining. By comparing a model trained on original human-written texts with one trained on simplified versions, we aim to isolate the specific role of text complexity.

In our experiments, both models train for a single epoch. The baseline model, `gpt2-hw`, processes about 2.34B tokens from human-written texts, while the simplified text model, `gpt2-simp`, is exposed to around 2.12B tokens. Additionally, human-written, domain-specific validation sets of roughly 23.4M tokens (about 5% of each domain) are evaluated every 300M tokens for regular checkpoints. Details on hyperparameter selection are provided in Appendix D. Pretraining for both models requires approximately 16 hours.

4.2.2 Domain Ablation Studies

A key aspect of our research examines whether text complexity's impact varies across content domains. The domain ablation experiments address this by systematically omitting one domain at a time and observing the effect on model performance. This approach is based on the idea that certain domains—such as legal or academic texts, which require a high degree of nuance—may rely more on complex linguistic structures, while other domains can effectively communicate core information even when simplified.

To investigate, we train 10 models—five on human-written texts and five on simplified texts. In

[5] https://huggingface.co/gpt2

each ablation run, one of the five domains is omitted, removing approximately 468M tokens from the training data. Pretraining for these ablation experiments takes around 13 hours per run, and the resulting models are fine-tuned on the SuperGLUE benchmark. This evaluation aims to determine whether omitting complex linguistic structures in specific domains differentially affects the model's general language understanding.

4.2.3 Simple-to-Complex Curriculum

Beyond directly comparing text complexity, we explore a two-phase pretraining strategy based on a simple-to-complex curriculum. We hypothesize that starting with simplified texts enables the model to quickly learn fundamental syntactic and semantic patterns, forming a foundation that is refined with later exposure to more intricate human-written texts.

To evaluate this, we compare two strategies. In the baseline, the model is trained for two epochs solely on the human-written corpus (roughly 4.68B tokens); we refer to this model as `gpt2-hw-2epoch`. The curriculum strategy trains on a concatenated corpus where the model first processes simplified texts and then transitions to human-written texts (roughly 4.46B tokens); we refer to this model as `gpt2-curriculum`. Validation loss is recorded every 600M tokens across domains, with seven intermediate checkpoints and a final model saved. Both runs require roughly 32 hours, and each checkpoint model is fine-tuned on SuperGLUE tasks. This approach tracks the evolution of language representations and determines whether early simplified pretraining provides lasting downstream benefits.

4.3 Downstream Tasks

To assess whether pretraining differences influenced by text complexity impact downstream performance, we fine-tune our pretrained models on the SuperGLUE benchmark (Wang et al., 2019), which offers a comprehensive suite for evaluating general language understanding. Our evaluation covers eight core tasks: BoolQ, CB, COPA, MultiRC, ReCoRD, RTE, WiC, and WSC.

For each task, we reformat the data into prompt-based inputs by appending the correct label and computing loss only on these label tokens. This ensures the model aligns its predictions with the desired output without being distracted by other tokens. During inference, candidate label tokens are appended to the prompt, and the candidate with the highest total log probability is selected (see Appendix E for examples).

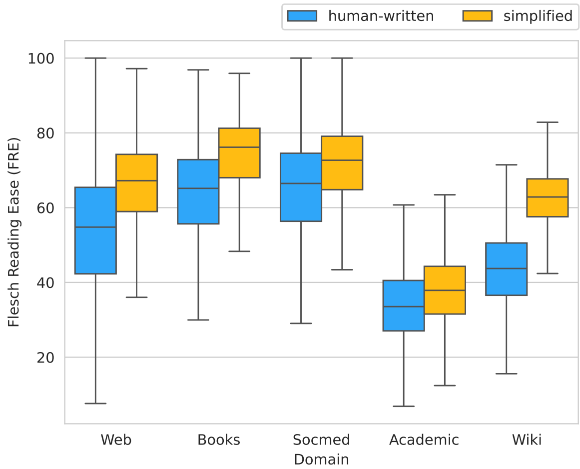

Figure 2: Flesh-Kincaid Reading Ease (FRE) scores of the human-written and simplified texts on each domain. Some documents fall outside the 0-100 range, so we clip them to 0 and 100 respectively.

The fine-tuning phase involves a per-task grid search for the best hyperparameters with a total combined runtime of approximately 26 hours per model. More details on hyperparameter selection, grid search, and final model selection are provided in Appendix D.

For evaluation, we use accuracy for 5 tasks (BoolQ, COPA, RTE, WiC, and WSC). For CB, MultiRC, and ReCoRD, we deviate from the official metrics since they do not reliably reflect performance in our setup. In CB, we report only accuracy—omitting F1, as predicting a single neutral label can boost F1 by over 11 points on a small, imbalanced dataset (16/250 in train, 5/56 in validation). For MultiRC, we report only micro F1 (equivalent to accuracy) and omit Exact Match (EM), which measures perfect passage-wise recall. For ReCoRD, we rely solely on EM, as token-overlap F1 can be inflated by partial matches. For transparency, we include additional results and analysis on the official metrics in Appendix G.

5 Results and Discussion

All results are from a single run only. For downstream performance, we report the best outcomes from a fixed hyperparameter grid. For reproducibility, we ensured that random seeds are properly set for all experiments.

	Avg.	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC
`gpt2-hw`	58.3	68.2	71.4	58.0	68.3	18.4	61.7	64.9	55.8
`gpt2-simp`	57.4	66.7	71.4	56.0	68.2	17.9	60.3	64.0	54.8
	(-0.9)	(-1.5)	(0.0)	(-2.0)	(-0.1)	(-0.5)	(-1.4)	(-0.9)	(-1.0)

Table 1: Comparison of `gpt2-hw` and `gpt2-simp` accuracy scores on the validation sets of eight SuperGLUE tasks. The **Avg.** column is the average of the eight task scores. The row below `gpt2-simp` shows the difference from `gpt2-hw` (green if higher, red if lower, gray if equal).

	Avg.	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC
`gpt2-hw`	56.5	68.5	74.0	46.6	64.0	17.8	58.4	62.4	60.3
`gpt2-simp`	54.7	66.9	69.6	47.8	63.9	17.9	54.4	61.4	55.5
	(-1.8)	(-1.6)	(-4.4)	(+1.2)	(-0.1)	(+0.1)	(-4.0)	(-1.0)	(-4.8)

Table 2: Comparison of `gpt2-hw` and `gpt2-simp` accuracy scores on the official test sets of eight SuperGLUE tasks. The **Avg.** column is the average of the eight task scores. The row below `gpt2-simp` shows the difference from `gpt2-hw` (green if higher, red if lower, gray if equal).

Corpus	Words	Types	TTR	Entropy
human-written	1.98B	7.98M	0.40%	10.75
simplified	1.83B	6.04M	0.33%	10.38

Table 3: Corpus statistics. Words are space-separated words, Types are unique word count, TTR is Type-Token Ratio, and Entropy refers to Unigram Entropy. Lower TTR means lower lexical diversity. Lower Entropy means lower complexity.

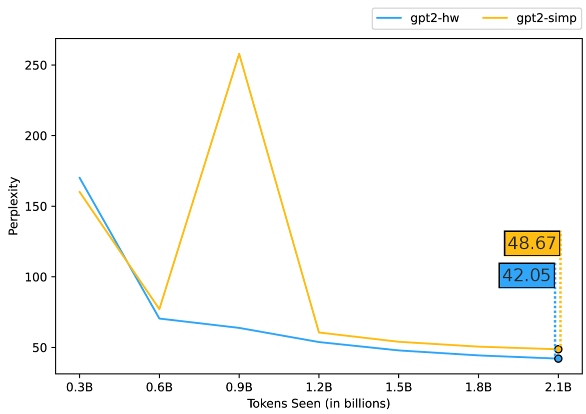

Figure 3: Perplexity vs. tokens seen graphs on the human-written validation set for both `gpt2-hw` and `gpt2-simp`. Perplexity is the exponentiation of loss and quantifies the model's "uncertainty."

5.1 Dataset Complexity Verification

Is our simplified text really simpler? To answer that question, we compute corpus corpus-level complexity metrics presented in Table 3 and document-level text complexity using the Flesch Reading Ease or FRE (Flesch, 1948). The simplified corpus has fewer words, lower Type-Token Ratio (TTR), and lower Unigram Entropy than its human-written counterpart which are all indicators of reduced complexity of simplified corpus.

For computing FRE, we use `py-readability-metrics`[6]. FRE considers text length, word count, and syllables per word, offering a rough complexity measure. A higher FRE implies simpler text (e.g., scores of 60 and above are considered easy; scores between 50 and 60 are fairly difficult; and scores below 50 are considered hard). While it doesn't capture all factors such as rare words or complex sentence structures, we use it for its practicality and simplicity.

Figure 2 shows that the FRE distribution of our simplified corpus is consistently higher than that of the human-written corpus across all domains. Some documents fall outside the 0–100 range, so we clip negative values to 0 and values above 100 to 100 (e.g., very long documents or texts with no punctuations). Notably, the academic and wiki domains are more complex than others.

[6]`https://github.com/cdimascio/py-readability-metrics`

5.2 Main Comparison: Human-Written vs. Simplified

5.2.1 Language Modeling Performance

To compare the relative language modeling performance of `gpt2-simp` with `gpt2-hw` in modeling human-written text, we compute the perplexity of both models on held-out **human-written** texts. Figure 3 shows that `gpt2-simp` exhibits comparable perplexity with `gpt2-hw`. The results are not surprising since a slight difference in the distribution between human-written and simplified texts is ex-

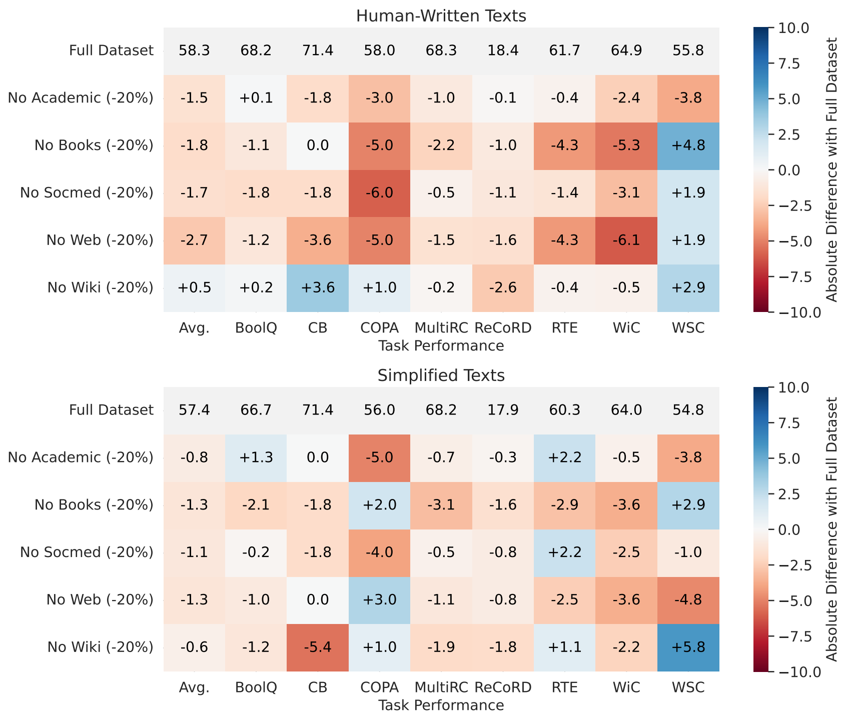

Figure 4: A heatmap of the differences on SuperGLUE task scores when removing one domain at a time from both the human-written and simplified datasets. Blue represents an increase in performance while red represents a decrease.

pected (e.g., stylistic differences and word choices). However, it is interesting to note that despite training solely on simplified texts, `gpt2-simp` was able to learn representations that can model human-written texts, comparable to `gpt2-hw`. These results suggest that the learned representations on simplified texts may be suitable for adaptation to human-written texts. For a detailed discussion on the spike in perplexity for `gpt2-simp` and domain-level perplexity, see Appendix F.

5.2.2 SuperGLUE Performance

Table 1 summarizes performance on the validation sets for eight SuperGLUE tasks. `gpt2-simp` achieves an average score of 57.4, just below the 58.3 of `gpt2-hw`. Most tasks show only slight differences between the models. Similarly, Table 2 shows that on the test set, `gpt2-simp` reaches an average of 54.7 compared to 56.5 for `gpt2-hw`, reflecting a very modest overall gap. While a few tasks even register small improvements, most differences remain minimal. These observations indicate that reducing linguistic complexity while keeping the core meaning intact has a limited effect on downstream performance.

5.3 Domain Ablation Results

Our domain ablation experiments (see Figure 4) systematically omit each domain from the training corpus in both human-written and simplified datasets, one at a time, to assess each domain's importance for downstream tasks under different linguistic conditions.

On the average SuperGLUE scores, omitting almost any domain slightly reduces performance. The primary exception is the wiki domain: removing it from the human-written dataset yields a modest improvement, while excluding it from the simplified dataset causes a small drop. In contrast, the other four domains incur greater losses when removed from human-written data compared to when they are removed from simplified data—seemingly more so for the academic and web domains—suggesting that complex, human-written text in these domains captures nuanced style and content better, whereas wiki text may be more effective in simplified form.

A detailed discussion on individual task effects is provided in Appendix H.

5.4 Curriculum Learning Effects

Figure 5 shows that `gpt2-curriculum` achieves overall lower perplexity on human-written texts

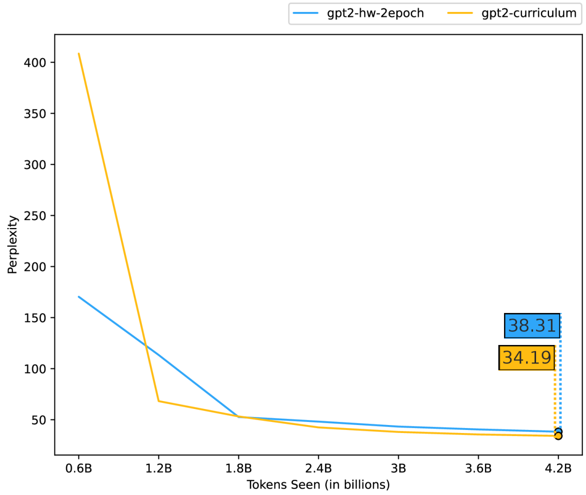

Figure 5: Perplexity on human-written validation set for both `gpt2-hw-2epoch` and `gpt2-curriculum`. `gpt2-curriculum` achieved lower perplexity on human-written text than the `gpt2-hw-2epoch` which was trained solely on human-written text.

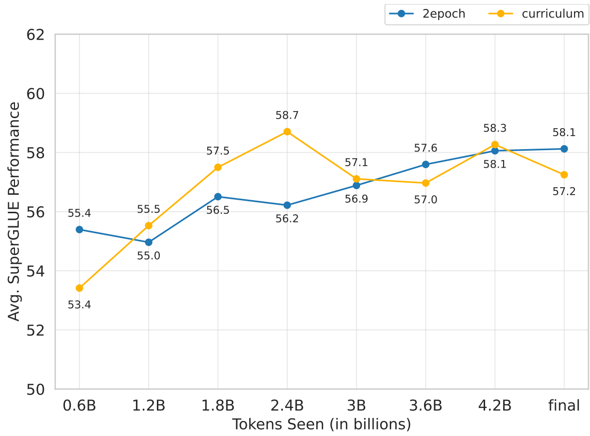

Figure 6: Average SuperGLUE score vs. number of tokens seen for both `gpt2-hw-2epoch` and `gpt2-curriculum`. Scores are obtained from the checkpoints of both models every 600M tokens seen.

compared to `gpt2-hw-2epoch`. We hypothesize that exposure to varied text versions, rather than repeated texts, enhances learning, similar to the findings of Allen-Zhu and Li (2024).

Figure 6 illustrates the average performance across all tasks, showing that `gpt2-curriculum` consistently achieves higher scores between 1200M and 3000M tokens. For a detailed breakdown of performance trends by task, see Appendix I.

The checkpoint experiments demonstrate that a curriculum training strategy, beginning with simplified texts and later transitioning to human-written texts, can accelerate early learning compared to the baseline model trained solely on human-written texts (`gpt2-hw-2epoch`). Although the early advantage of the curriculum approach eventually converges with the baseline, our findings indicate that it ultimately delivers performance on par with training exclusively on premium, human-written data, effectively replicating the long-term benefits of using only high-quality inputs.

6 Conclusion

In this work, we investigated the role of text complexity in the pretraining of language models, specifically examining whether simplified language, while preserving core information content, can yield representations that are as effective as those learned from more complex, human-written texts. Our experiments, which compared GPT2-small models pretrained on human-written versus simplified corpora, reveal that reducing lexical and syntactic complexity does not significantly impair downstream performance on a broad set of language understanding tasks such as those in the SuperGLUE benchmark. These findings suggest that, for the purposes of pretraining, the richness of information content is the primary driver of performance, rather than the complexity of the text form.

While our study is limited to the GPT2-small architecture and a specific experimental setting, the evidence presented here motivates future research into the interplay between text complexity, information content, and model performance across different architectures and larger-scale datasets.

Limitations

Our study has several limitations. First, the LLM-based simplification process can introduce inconsistencies in the information content due to the tendencies of LLMs to hallucinate. Second, the Flesch Reading Ease score only measures surface-level features and may not fully reflect deeper linguistic nuances. Third, our experiments are restricted to the GPT2-small architecture, so it is unclear how these findings extend to larger models with more parameters or different architectures. Fourth, our evaluation relies solely on the SuperGLUE benchmark, which might not capture all facets of language understanding, especially for more complex or generative tasks. Lastly, our domain ablation experiments cover only a subset of domains, limiting broader domain-specific insights.

References

Ameeta Agrawal and Suresh Singh. 2023. Corpus complexity matters in pretraining language models. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 257–263, Toronto, Canada (Hybrid). Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2023. Controlling pre-trained language models for grade-specific text simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819, Singapore. Association for Computational Linguistics.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. *Preprint*, arXiv:2309.14316.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy. Association for Computational Linguistics.

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

Savvas Chatzipanagiotidis, Maria Giagkou, and Detmar Meurers. 2021. Broad linguistic complexity analysis for greek readability classification. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–58.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Edgar Dale and Jeanne S Chall. 1949. The concept of readability. *Elementary English*, 26(1):19–26.

Vijeta Deshpande, Dan Pechi, Shree Thatte, Vladislav Lialin, and Anna Rumshisky. 2023. Honey, I shrunk the language: Language model behavior at reduced scale. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5298–5314, Toronto, Canada. Association for Computational Linguistics.

William H DuBay. 2004. The principles of readability. *Impact Information*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339, Torino, Italia. ELRA and ICCL.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh,

Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary

DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452, Marseille, France. European Language Resources Association.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012*, pages 1063–1080.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *Preprint*, arXiv:2412.05149.

Joseph Marvin Imperial. 2021. Bert embeddings for automatic readability assessment. *arXiv preprint arXiv:2106.07935*.

Joseph Marvin Imperial and Ethel Ong. 2021. A simple post-processing technique for improving readability assessment of texts using word mover's distance. *arXiv preprint arXiv:2103.07277*.

Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. Flesch or fumble? evaluating readability standard alignment of instruction-tuned language models. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 205–223, Singapore. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 207–217.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Bruce W. Lee and Jason Lee. 2023. Prompt-based learning for text readability assessment. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1819–1824, Dubrovnik, Croatia. Association for Computational Linguistics.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Changping Meng, Muhao Chen, Jie Mao, and Jennifer Neville. 2020. Readnet: A hierarchical transformer framework for web article readability analysis. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*, pages 33–49. Springer.

Sherin Muckatira, Vijeta Deshpande, Vladislav Lialin, and Anna Rumshisky. 2024. Emergent abilities in reduced-scale generative language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1242–1257, Mexico City, Mexico. Association for Computational Linguistics.

Yasuhiro Ozuru, Kyle Dempsey, and Danielle S McNamara. 2009. Prior knowledge, reading skill, and text cohesion in the comprehension of science texts. *Learning and instruction*, 19(3):228–242.

Francesca Padovani, Caterina Marchesi, Eleonora Pasqua, Martina Galletti, and Daniele Nardi. 2024. Automatic text simplification: A comparative study in Italian for children with language disorders. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 176–186, Rennes, France. LiU Electronic Press.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Donya Rooein, Paul Röttger, Anastassia Shaitarova, and Dirk Hovy. 2024. Beyond flesch-kincaid: Prompt-based metrics improve difficulty classification of educational texts. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 54–67, Mexico City, Mexico. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Horacio Saggion and Graeme Hirst. 2017. *Automatic text simplification*, volume 32. Springer.

Carolina Scarton, Gustavo Paetzold, and Lucia Specia. 2018. Text simplification from professionally produced corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Computational Processing of the Portuguese Language: 9th International Conference, PROPOR 2010, Porto Alegre, RS, Brazil, April 27-30, 2010. Proceedings 9*, pages 30–39. Springer.

Sean Trott and Pamela Rivière. 2024. Measuring and modifying the readability of English texts with GPT-4. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 126–134, Miami, Florida, USA. Association for Computational Linguistics.

Ciprian-Octavian Truică, Andrei-Ionuţ Stan, and Elena-Simona Apostol. 2023. Simplex: a lexical text simplification architecture. *Neural Computing and Applications*, 35(8):6265–6280.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

Xingmeng Zhao, Tongnian Wang, Sheri Osborn, and Anthony Rios. 2023. BabyStories: Can reinforcement learning teach baby language models to write better stories? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 186–197, Singapore. Association for Computational Linguistics.

A Manual selection of Dolma shards

For Dolma[7], We manually selected shards to reduce the total dataset size before we do any of our subsequent subsetting. We list below the specific shards (all are .json.gz) we used from Dolma:

```
books-0000, books-0001,
c4-0000, c4-0001,
pes2o_v2-0012,
reddit-v5-dedupe-pii-nsfw-toxic-0000,
reddit-v5-dedupe-pii-nsfw-toxic-0001,
reddit-v5-dedupe-pii-nsfw-toxic-0002
```

B Text Simplification Prompt

The prompt engineering is done through trial-and-error and judged by the authors according to the following qualitative criteria:

- Does it use simpler words? By "simpler words," we mean commonly used words.
- Does it convert compound or complex sentences into simple sentences?
- Does it preserve the original content and organization of thoughts?

Once we found a prompt that can reliably do all those things on a small sample, we used that prompt to transform the whole corpus.

The final prompt is shown below:

—

Role Description: You are an experienced educator and linguist specializing in simplifying complex texts without losing any key information or changing the content. Your focus is to make texts more accessible and readable for primary and secondary school students, ensuring that the essential information is preserved while the language and structure are adapted for easier comprehension.

—

Task Instructions: 1. Read the Following Text Carefully: - Thoroughly understand the content, context, and purpose of the text to ensure all key information is retained in the simplified version.

2. Simplify the Text for Primary/Secondary School Students: - Rewrite the text to make it more accessible and easier to understand. - Use age-appropriate language and simpler sentence structures. - Maintain all key information and do not omit any essential details. - Ensure that the original meaning and intent of the text remain unchanged.

3. Preserve Key Information: - Identify all essential points, facts, and ideas in the original text. - Ensure these elements are clearly presented in the simplified version.

4. Avoid Adding Personal Opinions or Interpretations: - Do not introduce new information or personal views. - Focus solely on simplifying the original content.

—

Simplification Guidelines:

Sentence Structure: - Use simple or compound sentences. - Break down long or complex sentences into shorter ones. - Ensure each sentence conveys a clear idea.

Vocabulary: - Use common words familiar to primary and secondary school students. - Replace advanced or technical terms with simpler synonyms or provide brief explanations. - Avoid jargon unless it is essential, and explain it if used.

Clarity and Coherence: - Organize the text logically with clear paragraphs. - Use transitional words to connect ideas smoothly. - Ensure pronouns clearly refer to the correct nouns to avoid confu-

[7] https://huggingface.co/datasets/allenai/dolma

sion. - Eliminate redundancies and unnecessary repetitions.

Tone and Style: - Maintain a neutral and informative tone. - Avoid overly formal language. - Write in the third person unless the text requires otherwise.

—

Output Format: Provide the simplified text in clear, well-organized paragraphs. Do not include the original text in your output. Do not add any additional commentary or notes. Ensure the final output is free of grammatical errors and is easy to read. Output <|eot$_i d| >$ $rightafterthesimplifiedtext.$

—

Example Simplifications:

Example 1:

Original Text: "Photosynthesis is the process by which green plants and some other organisms use sunlight to synthesize foods from carbon dioxide and water. Photosynthesis in plants generally involves the green pigment chlorophyll and generates oxygen as a byproduct."

Simplified Text: "Photosynthesis is how green plants make food using sunlight, carbon dioxide, and water. They use a green substance called chlorophyll, and the process produces oxygen.<|eot$_i d| >$ ”

Example 2:

Original Text: "Global warming refers to the long-term rise in the average temperature of the Earth’s climate system, an aspect of climate change shown by temperature measurements and by multiple effects of the warming."

Simplified Text: "Global warming means the Earth’s average temperature is increasing over a long time. This is part of climate change and is shown by temperature records and various effects.<|eot$_i d| >$ ”

Example 3:

Original Text: "The mitochondrion, often referred to as the powerhouse of the cell, is a double-membrane-bound organelle found in most eukaryotic organisms, responsible for the biochemical processes of respiration and energy production through the generation of adenosine triphosphate (ATP)."

Simplified Text: "A mitochondrion is a part of most cells that acts like a powerhouse. It has two membranes and makes energy for the cell by producing something called ATP.<|eot$_i d| >$ ”

—

Text to Simplify: <Insert Text Here>

—

Your Output:

C Skipping or Rejecting Simplification

We choose to skip or reject the simplification step under the following conditions: (1) the paragraph is too short relative to its full document; (2) the paragraph is too long; or (3) the transformation is significantly shorter or longer than the original text.

Condition (1) is based on two key observations. First, some textual artifacts, like titles and author names, don’t require simplification. Second, very short inputs often trigger text completion instead of simplification. For example, the input **"MAHATMA GANDHI"** generates a passage about the person rather than a simplified version. To handle such cases, we use heuristics to determine whether a document or paragraph should be skipped. First, we apply a hard rule: a document is skipped if there is only one paragraph or the minimum paragraph length is greater than or equal to the standard deviation of paragraph token counts within a document. Otherwise, each paragraph in the document is evaluated based on two criteria: it is skipped if it contains **10 or fewer space-separated words** or if its **GPT-2 token count falls below the quantile threshold**. The quantile threshold varies by domain (e.g., **0.25 for books, 0.15 for others**). For example, for the books domain, the quantile threshold is 0.25 (25th percentile), meaning paragraphs with token counts below the 25th percentile will be skipped.

Condition (2) is based on the observation that paragraphs exceeding **1,500 tokens** tend to be structured texts like tables, name lists, or tables of contents, which do not need simplification. To handle such cases, we simply skip the paragraph if it exceeds 1,500 tokens. While quantile heuristics could be used, we chose the simpler heuristic.

Condition (3) is motivated by two observations. First, we observed that when asked to simplify a long input, the model tends to summarize it, significantly shortening the text and losing its original structure. Second, the model sometimes appends extra text, such as explanations after the answer. To detect cases where the output is too short or too long relative to the source, we compute the document length ratio (output_length/source_length) and reject outputs with a ratio below 0.5 or above 1.5 (i.e. a change of more than 50%), reverting to the original paragraph.

D Training Hyperparameters

For pretraining all of our models, to ensure smooth convergence, we employ a warmup ratio of 5% alongside a linear learning rate scheduler. The effective batch size is set to 384, achieved by running a batch size of 4 per GPU across 8 GPUs with 12 gradient accumulation steps. A preliminary two-stage learning rate sweep on 10% of the human-written corpus helped us determine a final learning rate of 6e-4.

The experimental configuration for finetuning on SuperGLUE tasks varies per task, depending on dataset size: for smaller tasks such as CB, COPA, RTE, WiC, and WSC, we use an effective batch size of 8 (distributed as one per GPU on 8 GPUs), whereas for larger datasets like BoolQ, MultiRC, and ReCoRD, an effective batch size of 32 (4 per GPU on 8 GPUs) is utilized. For all tasks, we perform a grid search over 1–2 epochs, exploring learning rates ranging from 2e-6 to 1e-4, and select the optimal hyperparameters for each pretrained model based on their highest macro F1 score on the validation sets. The use of macro F1 is particularly crucial as it offers a more balanced evaluation in scenarios where class imbalance might otherwise skew accuracy metrics; in the worst case, we found models collapsing to only predicting a single label for the entire dataset, indicating too much bias towards the tokens for one of the labels. We therefore avoid selecting a model that exhibits such imbalanced prediction strategies. We include the final macro F1 scores for `gpt2-hw` and `gpt2-simp` in Table 5.

E SuperGLUE Prompts

The following illustrate our prompt structures for each of the 8 SuperGLUE tasks:

For BoolQ, a question is paired with a passage, and the binary answer is appended:

> **Question**: Is water wet?
>
> **Passage**: Water is a liquid at room temperature with cohesive properties.
>
> **Answer**: Yes

For CB, a premise and a hypothesis are provided, followed by a label indicating their relationship:

> **Premise**: The new policy will reduce emissions.
>
> **Hypothesis**: The policy is effective in reducing emissions.
>
> **Label**: Contradiction

For COPA, a premise, a question, and two choices are presented; the answer indicates the most plausible outcome:

> **Premise**: Sarah forgot her umbrella.
>
> **Question**: What is the most likely outcome?
>
> **Choice 1**: She got wet in the rain.
>
> **Choice 2**: She stayed dry. Answer: 2

For MultiRC, each candidate answer is treated as a separate entry, and the model classifies its correctness:

> **Passage**: The experiment showed a significant increase in reaction times.
>
> **Question**: Did the reaction times increase?
>
> **Candidate Answer**: Yes, they did.
>
> **Is this answer correct?** Yes

For ReCoRD, the passage is first cleaned by removing any `@highlight` tokens. The query is then truncated at the `@placeholder` (removing it and all subsequent text), and concatenated with the cleaned passage. The gold answer is appended so that the model learns next-token prediction for the missing entity:

> In the heart of the desert, ancient ruins spoke of a lost civilization. A recent discovery suggests that Remnants

For RTE, a premise and a hypothesis are provided with a label indicating entailment:

> **Premise**: The cat sat on the mat.
>
> **Hypothesis**: A cat is resting on a mat.
>
> **Label**: Entailment

For WiC, a target word is given along with two sentences, and the task is to determine if the word's meaning is the same in both:

Word: bank

Sentence 1: I sat on the river bank.

Sentence 2: I deposited money at the bank.

Same meaning? No

For WSC, a sentence is provided that requires resolving a pronoun reference:

Text: The trophy didn't fit in the brown suitcase because it was too large.

Is the reference correct? Yes

F Perplexity Spike and Domain-wise Perplexity

The spikes in the validation perplexity of `gpt2-simp` is due to the instabilities during pretraining. Figure 8 shows the training loss for both models. Note that in both setups, the spikes occurred at around the same time. However, it didn't show a spike for `gpt2-hw` because the checkpoint validation occurred before the spike, and by the time the next checkpoint was reached, `gpt2-hw` had already recovered. Our hypothesis is that there must have been very bad batches of data at those steps which caused the model to diverge. However, we continued the training since the model ended up recovering in later steps.

The domain-wise perplexity of `gpt2-hw` and `gpt2-simp` is presented at Figure 7. `gpt2-simp` exhibits perplexity comparable to `gpt2-hw`, differing by 6 to 9 points across all domains.

G Official SuperGLUE Results

Table 4 showcases the official results obtained from the online submission portal of SuperGLUE. `gpt2-simp` scores 50.3, only 2.2 lower than `gpt2-hw`, which scores 52.5.

H Domain Ablation Results

Examining the results for each individual task in our domain ablations (see Figure 4) reveals further subtleties. COPA and RTE show particularly strong sensitivity to domain removal, and in opposite ways for human-written vs. simplified datasets. For COPA, excluding books or web from the human-written corpus reduces accuracy by up to 5 points, but excluding these same domains from the simplified corpus actually improves accuracy by 2-3 points. A likely explanation is that COPA scenarios are often grounded in nuanced, real-world contexts that the human-written books domain captures better than its simplified counterpart. For example:

Premise: "The host cancelled the party."

Choice 1: "She was certain she had the flu."

Choice 2: "She worried she would catch the flu."

Label: "Choice 1"

By contrast, RTE also suffers large losses from excluding the books and web domains in the human-written corpus, yet still sees small drops when those domains are removed from the simplified corpus. Meanwhile, removing the academic, social media, or wiki domains from the human-written dataset causes only minor performance decreases, whereas omitting them from the simplified dataset actually produces moderate gains. This pattern suggests that, for tasks like RTE requiring more complex reading comprehension, the simplified versions of certain domains (e.g., academic or wiki) may not convey the linguistic subtleties well enough. For example:

Premise: "It rewrites the rules of global trade, established by the General Agreement on Tariffs and Trade, or GATT, in 1947, and modified in multiple rounds of negotiations since then."

Hypothesis: "GATT was formed in 1947."

Label: "Not Entailment"

Overall, these findings show that even seemingly small shifts in domain coverage can have task-specific consequences, and that the linguistic complexity of the text in a domain may be critical, not only for accurately capturing the nuances in the content, but also for developing the linguistic foundations appropriate for certain downstream tasks. Maintaining diversity in pretraining data, while also aligning text complexity to the needs of each target task, appears to be key in optimizing performance.

I Curriculum Experiment Results

This appendix contains a more detailed discussion on the task-by-task performance of `gpt2-hw-2epoch` and `gpt2-curriculum` every 600M tokens seen.

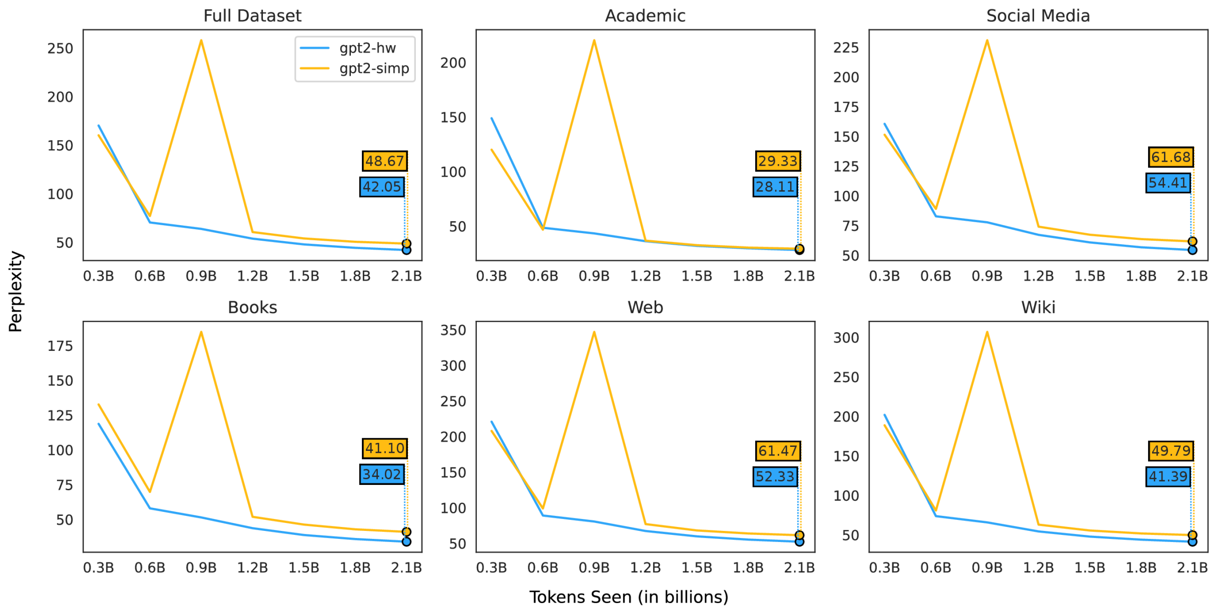

Figure 7: Domain-wise perplexity vs. tokens seen graphs on the human-written validation set for both `gpt2-hw` and `gpt2-simp`.

	Avg.	BoolQ Acc.	CB F1 / Acc.	COPA Acc.	MultiRC F1$_a$ / EM	ReCoRD F1 / EM	RTE Acc.	WiC Acc.	WSC Acc.
`gpt2-hw`	52.5	68.5	59.8 / 74.0	46.6	64.0 / 14.7	18.1 / 17.8	58.4	62.4	60.3
`gpt2-simp`	50.3	66.9	47.9 / 69.6	47.8	63.9 / 14.7	18.2 / 17.9	54.4	61.4	55.5
	(-2.2)	(-1.6)	(-11.9 / -4.4)	(+1.2)	(-0.1 / 0.0)	(+0.1 / +0.1)	(-4.0)	(-1.0)	(-4.8)

Table 4: Comparison of `gpt2-hw` vs. `gpt2-simp` scores on the official test set metrics on the eight SuperGLUE tasks. For BoolQ, COPA, RTE, WiC, and WSC the metric is Accuracy; for CB the metrics are F1 / Accuracy; for MultiRC the metrics are F1$_a$ / EM; for ReCoRD the metrics are F1 / Accuracy. The **Avg.** column indicates the overall score. The row below the Simplified scores shows the difference from Baseline (green if higher, red if lower, gray if equal).

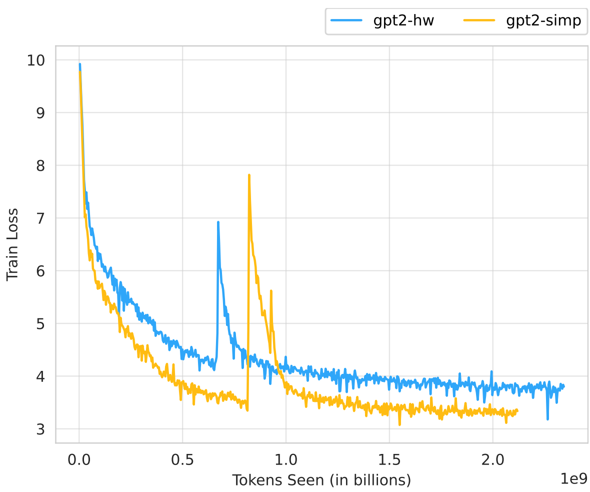

Figure 8: Training loss of `gpt2-hw-2epoch` and `gpt2-curriculum` exhibits spikes at around the same time.

As depicted in Figure 9, which presents eight subplots corresponding to each SuperGLUE task, the curriculum model (`gpt2-curriculum`) shows clear upward trends on tasks such as BoolQ, RTE, WiC, and MultiRC. Between the 1200M and 2400M token checkpoints, `gpt2-curriculum`'s performance even marginally surpasses that of `gpt2-hw-2epoch` on said tasks, demonstrating the early advantages of a simple-to-complex training approach. Moreover, the final `gpt2-curriculum` slightly outperforms the final `gpt2-hw-2epoch` on five tasks (BoolQ, CB, MultiRC, RTE, and WSC).

A plausible explanation for these trends is that the initial exposure to simplified texts enables the model to more easily acquire essential syntactic and semantic patterns, thereby establishing a stronger linguistic foundation early on.

In contrast, on the ReCoRD task, `gpt2-hw-2epoch` consistently outperforms `gpt2-curriculum` at every checkpoint. Notably,

	Avg.	BoolQ F1	CB F1	COPA F1	MultiRC F1	ReCoRD -	RTE F1	WiC F1	WSC F1
gpt2-hw	60.0	65.1	60.2	50.9	68.0	-	60.0	64.4	51.1
gpt2-simp	57.6 (-2.4)	62.8 (-2.3)	49.8 (-10.4)	51.6 (+0.7)	68.0 (0.0)	-	56.8 (-3.2)	63.4 (-1.0)	51.0 (-0.1)

Table 5: Comparison of `gpt2-hw` vs. `gpt2-simp` macro F1 scores on 7 out of 8 SuperGLUE task validation sets. No values are included for ReCoRD since it is not a fixed-choice task.

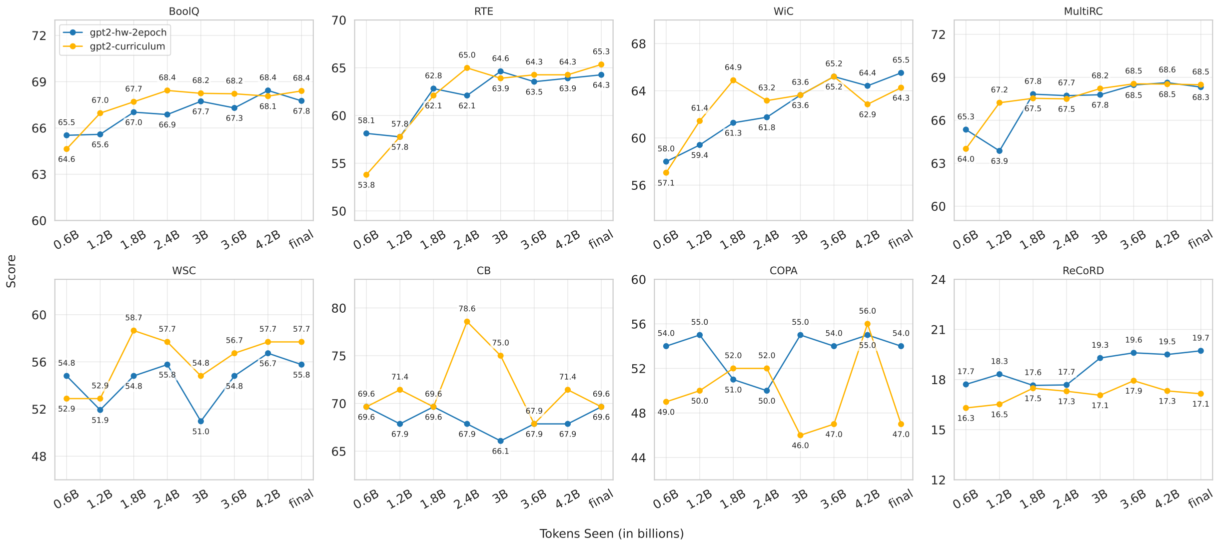

Figure 9: Subplots for SuperGLUE task scores vs. number of tokens seen on each task for both `gpt2-hw-2epoch` and `gpt2-curriculum`. Scores are obtained from the checkpoints of both models every 600M tokens seen.

however, both models show uniformly poor performance on ReCoRD, with scores ranging only between 16 and 20, compared to most other tasks that fall between 50 and 80. Possible reasons for these low ReCoRD scores include the inherent difficulty of the task, the GPT2-small architecture's limited capacity, and the mismatch between ReCoRD's advanced reading-comprehension style and a next-token prediction paradigm.

It is important to note, however, that the average performance curve of `gpt2-curriculum` exhibits a spike at the 2400M token checkpoint, driven predominantly by an anomalously high score on CB. Additionally, performance on CB and COPA appear erratic for both models, without a clear trend of improvement as pretraining continues. This instability is likely due to the inherent sensitivity of their small datasets to statistical noise, random data sampling variations, and potential overfitting, being only a few hundred instances each.

Overall, these findings suggest that a simple-to-complex curriculum provides a beneficial "warm-up" phase for many language understanding tasks.