WhatIf: Leveraging Word Vectors for Small-Scale Data Augmentation

Alex Lyman and **Bryce Hepner**
Brigham Young University
alexlyman@byu.edu

Abstract

We introduce **WhatIf**, a lightly supervised data augmentation technique that leverages word vectors to enhance training data for small-scale language models. Inspired by reading prediction strategies used in education, **WhatIf** creates new samples by substituting semantically similar words in the training data. We evaluate **WhatIf** on multiple datasets, demonstrating small but consistent improvements in downstream evaluation compared to baseline models. Finally, we compare **WhatIf** to other small-scale data augmentation techniques and find that it provides comparable quantitative results at a potential tradeoff to qualitative evaluation.

1 Introduction

The use of Large Language Models (LLMs) has exploded in the recent past, with LLMs becoming the state of the art for most NLP tasks. While statistical models of language have been around for decades (Markov, 2006), the introduction of the Transformer (Vaswani, 2017) set the stage for a new era in language modeling.

Early Transformer-based language models such as BERT (Devlin, 2018) and GPT (Radford, 2018) are very small by today's standards, with a few hundred million parameters each. In the intervening years, models have grown exponentially both in number of parameters and number of training tokens. These increases in size have been accompanied by increases in performance, with abilities emerging as a consequence of model scale (Wei et al., 2022). Current state of the art models tend to have tens of billions to hundreds of billions of parameters and are trained on trillions of training tokens.

In this paradigm of increasing scale, there has been relatively little focus on small-scale language modeling, which tends to be restricted to domains such as low-resource machine translation.

The BabyLM challenge (Choshen et al., 2024) seeks to focus researchers on very small-scale language modeling. The challenge involves using either a 10 or 100 million word "developmentally plausible" corpus (Warstadt et al., 2023), with 100 million words being roughly amount of words a child hears before reaching adulthood. Working at this small scale enables researchers to focus on cognitively inspired methods of language modeling as well as to iterate on language modeling experiments, which is impractical at 100-billion parameter scales.

While much recent focus on language modeling has involved scaling up parameter and training token counts, these approaches have drawbacks, including environmental concerns and inaccessibility of hardware (Bender et al., 2021). As a consequence, there has been a recent focus on mid-scale language modeling, creating models that can be run locally on devices such as consumer PCs or smartphones. This research has been promising. Microsoft's Phi models (Li et al., 2023; Abdin et al., 2024) boast impressive performance on many language modeling benchmarks, in spite of having only a few billion parameters. Phi's major innovation is using only "textbook quality data", curated from only high-quality sources rather than semi-filtered data of dubious quality scraped from the internet.

At a much smaller parameter scale, Eldan and Li (2023) trained very small transformers on TinyStories, a synthetic dataset of children's stories. In spite of parameter counts below 10 million, these tiny models were able to generate coherent text with real world knowledge and logic.

The trend towards improving data quality and quantity rather than solely scaling model parameters has also been applied successfully to larger-scale language modeling. Llama 3 (Team, 2024) attributes its significant improvements in performance over Llama 2 (Touvron et al., 2023) not to

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 229–236
November 15-16, 2024 ©2024 Association for Computational Linguistics

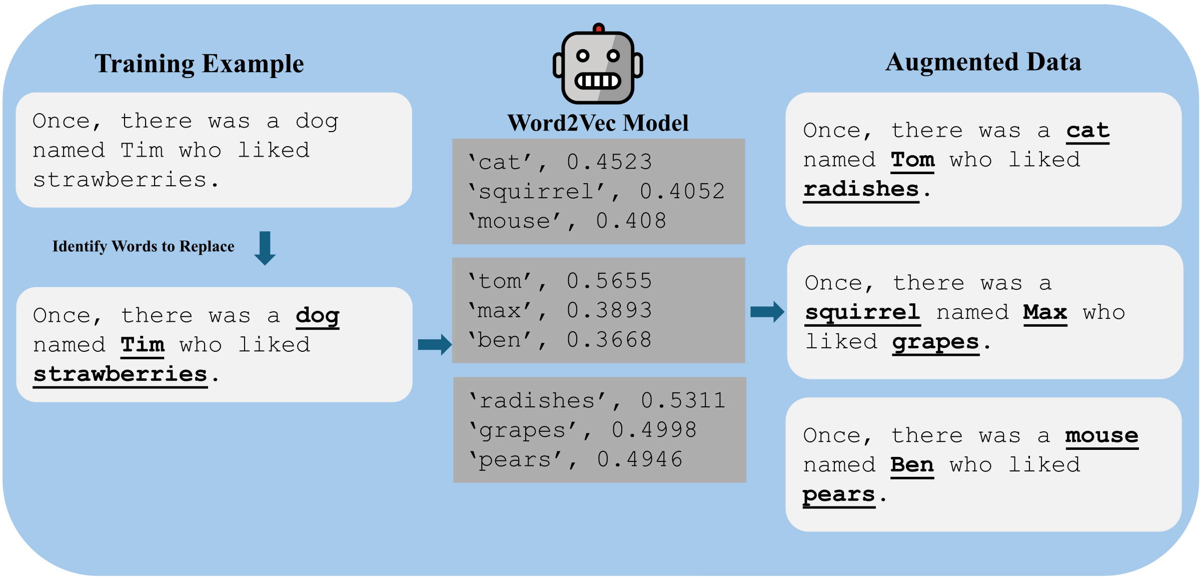
Training Example
Once, there was a dog named Tim who liked strawberries.
Identify Words to Replace
Once, there was a dog named Tim who liked strawberries.
Word2Vec Model
'cat', 0.4523
'squirrel', 0.4052
'mouse', 0.408
'tom', 0.5655
'max', 0.3893
'ben', 0.3668
'radishes', 0.5311
'grapes', 0.4998
'pears', 0.4946
Augmented Data
Once, there was a cat named Tom who liked radishes.
Once, there was a squirrel named Max who liked grapes.
Once, there was a mouse named Ben who liked pears.

Figure 1: Illustration of the data augmentation technique.

changes in architecture, but to "improvements in data quality and diversity as well as by increased training scale." Using higher quality input, Llama 3 was trained on roughly ten times as many tokens as Llama 2.

Our research is motivated by both the language modeling research on training data, as well as children's processes of language acquisition.

Training on more data has a human analog. One of the strongest predictors of children's linguistic development is the amount and type of language they hear (Weisleder and Fernald, 2013). Children who hear more words tend to have larger vocabularies, which correlates with better educational outcomes later in life (Hart et al., 1997; Hoff, 2003).

We introduce **WhatIf**, a lightly supervised data augmentation technique that uses word vectors to augment training data. **WhatIf** substitutes words in the training corpus for semantically similar words, enabling our baby models to consider novel yet similar text to the training data.

WhatIf is inspired by a method of improving reading comprehension called predicting. With this strategy, teachers instruct students to periodically ask questions about the text. These questions can be predictions about what might occur later in the text or counterfactuals, which usually take the form of "What If?" questions. Teaching with prediction strategies improves reading instruction outcomes. Both children (Küçükoğlu, 2013) and second language learners (Ali and Razali, 2019) show improved reading comprehension when employing prediction strategies.

Our method is lightly supervised, and requires a small part-of-speech dictionary, which we count as part of our token budget. This too has an analog in real-world language acquisition. Children receive explicit grammatical knowledge. For example, children who produce ungrammatical speech are often corrected by a parent or caregiver. Children who receive explicit grammatical instruction and have explicit grammatical awareness tend to develop better linguistic skills (Ehri et al., 2001).

We perform experiments and show that **WhatIf** increases model performance on a variety of evaluation tasks, and performs comparably to other small-scale Language Model data augmentation techniques, though these quantitative gains come at a cost to text quality.

2 Methods

2.1 Data Augmentation

The core of the data augmentation technique is word vectors. We use the Word2Vec algorithm (Mikolov et al., 2013) to create semantic embeddings for each word in our training corpus. When trained over a sufficiently large corpus, Word2Vec embeddings cause similar words to end up with similar vector representations in the high-dimensional space. If the words are sufficiently semantically similar, changing one word for its nearest neighbor should preserve most of the sense and meaning of the text, while still creating novel, useful training examples.

We first split the corpus into sentences, then use the sentences to train a Word2Vec model. Then for each training example, we select p percent of the content words at random, excluding function words, which do not have grammatical equivalents. For each of the chosen words, we use the word vector model to select the nearest neighbor via cosine similarity, which is most semantically similar. Then, we check whether both the word and its candidate replacement have the same part of speech. If so, we replace each occurrence of the word in the training example with the candidate replacement. Otherwise, we repeat with the next nearest neighbor until we select a viable candidate or reach a preset distance threshold in vector space. The use of a threshold prevents the selection of semantically distinct words that happen to be nth near neighbors.

Once a viable replacement is selected, each occurrence of the word in the training example is replaced. This guarantees semantic continuity throughout the training example. This process can be repeated any number of times to increase the amount of data available to the model, starting from the gold standard each time. At each iteration the word vector model selects less similar words, theoretically enabling us to create large amounts of data of decreasing quality.

This technique does not guarantee grammatical or correct training examples. For example, according to the word vector model trained on our TinyStories corpus, the most similar word to *old* is *elderly*.

This works in contexts like:

*The **old** man / the **elderly** man.* ✓

However, this causes problems in contexts such as:

*The **old** castle / the **elderly** castle.* X

2.2 Model and Training Details

Because our primary focus is on training data, we select a simple model for our experiments based on the GPT-2 architecture (Radford et al., 2019). Although previous competition results show that GPT-2 is not the best architecture for small-scale language modeling, we choose it for ease of use, familiarity, and reproducibility.

We use the same training setup for all models, a version of the GPT-2 Small checkpoint with reduced size. While GPT-2 Small has an inner dimension of 768, we halve that size for an inner dimension of 384. We also halve the context window size from 512 to 256. Our models each have 26,000,640 trainable parameters.

To be able to iterate over many experiments, our models are optimized to train quickly, with a potential tradeoff in absolute performance. This is achieved both by reducing the size of the models and training them in FP16 precision.

To maximize training speed, all models were trained across 8 NVIDIA A100 GPUs with a batch size of 16, with a torch manual seed set to the same value for each model.

Across all experiments, batches are shuffled between each epoch.

We train a tokenizer on each dataset using HuggingFace's BPE implementation [1], with a vocabulary size of 12000.

Because different data augmentation techniques result in training sets of different sizes, we checkpoint and evaluate using steps instead of epochs. We train all models for roughly the equivalent of 100 epochs of the un-augmented dataset, and evaluate every 5,000 steps. Model performance tends to peak between 10,000 and 25,000 steps. (20-50 non-augmented epochs) To maintain consistency, we evaluate the 25,000 step checkpoint of each model for final evaluation.

2.3 Dataset Details

To ensure our results are not dataset dependent, we perform all experiments on two datasets. The first is a lightly filtered version of the BabyLM 2024 Strict-Small dataset comprising roughly 9,300,000 tokens. This data includes transcribed speech, narrative, and instructional texts. The second is a subset of the TinyStories dataset with roughly 9,950,000 tokens. These stories are synthetic data generated by GPT-3.5 and GPT-4, and are all short narratives with a target audience of 3-year-old children.

For each dataset, we use a small portion of the token budget for a part-of-speech dictionary. TinyStories has 12,233 key-value pairs for 24,466 total tokens, and BabyLM has 128,124 key-value pairs for 256,248 tokens. The part-of-speech dictionaries use Penn Treebank P.O.S. tags (Marcus et al., 1993). We also use the 159-word list of English stopwords from the NLTK package (Bird et al., 2009). In both cases, the count of training words and data augmentation materials falls under the 10 million token budget.

[1] https://github.com/huggingface/tokenizers

Models are trained on the training samples, with the dictionaries being used only for the data augmentation process.

2.4 Data Pre-Processing

Natural language occurs in context. In initial experiments, we found that joining lines from the BabyLM dataset into chunks led to large gains over to passing training examples line-by-line. Using contextual chunks enables the model to learn features of natural language such as conversational turn-taking. Across subsets of the training data, lines vary wildly in size. For each subset of the corpus we join a different number of lines to create each training example, with the goal of creating chunks of around 150 words. The 150 word mark was chosen because it enables most tokenized examples to fit within the 256 token context window. It is also close to the average number of words in the stories from the TinyStories dataset, allowing for an apples-to-apples comparison between models trained on both datasets.

3 Results

We perform a variety of experiments to probe the efficacy of our data augmentation technique. All experiments are performed on both the TinyStories dataset and the BabyLM Strict-Small dataset.

For our baselines, we train models using a standard language modeling approach. These examples occasionally need to be truncated, but thanks to the data pre-processing, the overwhelming majority of samples do not require truncation.

Because each pass of the augmentation process results in lower quality data, we experiment with how many passes of augmented data we create, $n = 5$ or $n = 10$ passes. For every n passes of augmented data, we also include one pass of the non-augmented gold standard data. This means only $\frac{1}{6}$ or $\frac{1}{11}$ of data seen while training on augmented data is gold-standard data. We also experiment with the percent of content words to replace, either 50%, which leaves the sample recognizable, or 100%, which drastically changes the training example. Examples of different degrees of augmentation can be found in the appendix.

3.1 Quantitative Results

We evaluate our models using the competition's default evaluation harness (Gao et al., 2023) and metrics: BLIMP (Warstadt et al., 2020), EWOK (Ivanova et al., 2024), and GLUE.

As shown in Table 1, **WhatIf** provides a small but consistent gain of 1 to 2 percentage points over the baselines.

Interestingly, benchmarks offer no clear trend as to the ideal hyperparameters for the data augmentation technique. The 5 pass models usually outperform their 10 pass counterparts, but by such a small margin that no clear conclusion can be drawn. While all augmented models outperform the baseline, there is not a clear winner.

We compare our results with a variant of the Contextualizer (Xiao et al., 2023), one of the best-performing data augmentation methods from the 2023 challenge. In our variant, **Contextualizer-like**, before each training pass we tokenize the whole dataset and shuffle the training examples. We then concatenate the tokenized samples and break them into 256-token chunks. We find that **Contextualizer-like** performs at a similar rate to **WhatIf** with a 1-2 percentage point increase over the pad and truncate baseline.

Finally, we ensemble our data augmentation technique with the **Contextualizer-like** algorithm to see if combining the methods causes an additional gain in performance. The results show that while both **WhatIf** and **Contextualizer-like** provide gains in performance, ensembling the two of them does not provide additional benefit.

3.2 Qualitative Results

Although **WhatIf** produces a small quantitative improvement as measured by benchmarks, models trained on augmented data can produce qualitatively worse text. To demonstrate this, we generate short completions to the prompt `Once upon a time` with three of our models trained on the TinyStories dataset with varying degrees of augmentation: the baseline model, the 5-pass-replace-50% model, and the 10-pass-replace-100% model. Samples are generated using top-k sampling with a temperature of 1 and a k of 20.

Baseline Model

```
Once upon a time, in a big forest,
there was a little bird.  The little
bird lived in a cage.  The bird had
a mommy bird.  The mommy bird could
not see the little bird in the cage.
The mommy bird was sad...
```

The baseline model generates a reasonable narrative, comparable with the output from the original TinyStories paper.

		BLIMP	BLIMP Sup.	EWOK	GLUE	Average
TinyStories	Baseline	55.9	52.9	51.3	59.6	54.925
	5 Pass Replace 50%	58.8	57.1	50.6	59.8	**56.575**
	10 Pass Replace 50%	58.8	54.2	50.9	60.6	56.125
	5 Pass Replace 100%	59.2	54.9	50	61.4	56.375
	10 Pass Replace 100%	58.3	54.1	50.9	60.7	56
	Contextualizer-like	59.1	54	51.5	60.1	56.175
BabyLM	Baseline	63.7	54.6	49.7	60.5	57.125
	5 Pass Replace 50%	64.5	56.6	50.8	60.9	58.2
	10 Pass Replace 50%	63.9	56.5	50.6	60	57.75
	5 Pass Replace 100%	66.3	59	50.6	60.4	**59.075**
	10 Pass Replace 100%	64	56.1	51.2	60.8	58.025
	Contextualizer-like	66.9	56.4	51.7	60.7	58.925

Table 1: Results of baseline and augmented models, evaluated at the 25,000 step checkpoint.

10-pass-replace-100%

```
Once upon a time, there was a child
named True.  True started to travel
with his brother, Bob.  They were
very stupid at riding games.  One
day, True returned hurt while they
worked.  Bob felt confused.  He said
to True, "I am sorry, let's travel
to my parent....
```

This text is lower in quality. We see examples of grammatical constructions that make no semantic sense, `"very stupid at riding games"` as well as poor world knowledge, e.g. `True` is not a normal name.

5-pass-replace-50%

```
Once upon a time, there was a little
kitten named Amy.  Amy liked to cook
with her mom.  One day, they decided
to cook a big salad for lunch.  Amy
was very happy.  Amy's mom told her,
"Amy, can you put the salad in the
oven?" Amy opened the oven and put
the salad in the oven...
```

While the overall story lacks some world knowledge (salad is not typically cooked in an oven), this output suggests this somewhat augmented training mix may be a reasonable compromise between quantity and quality, though further experiments are necessary to identify the ideal training mixture.

4 Discussion

Just like children, small-scale language models benefit from additional data. **WhatIf** shows mild but consistent benchmark improvement above the baseline across datasets.

We expect this to improves performance on benchmarks by exposing the LM to new scenarios during training. In practice, the augmented data is sometimes fairly low-quality. As a consequence, the LM can learn incorrect facts about the world. For example, augmentation may replace the word *Mom* with *Dad*, without replacing gendered pronouns *she* with *he*. This does not seem to have a large negative effect on the model's grammatical abilities, since BLIMP and BLIMP supplement scores improve with **WhatIf** augmentation. However, EWOK scores do not improve decreasing slightly when applied to the TinyStories dataset. We suspect that the LM earns incorrect information about the world from bad correlations in the augmented data.

The fact that both **WhatIf** and **Contextualizer-like** provide similar gains suggests that manipulating the training data in some well-informed way provides modest performance gains. Since we observe diminishing returns when ensembling both methods, this might mean that both methods are acting on a similar axis to make the training data more useful to the model.

5 Limitations and Future Work

This work is only a partial realization of the underlying idea that data augmentation with word vectors could improve model performance. We suspect that small changes to the data augmentation algorithm could bear significant fruit. An additional round of validation to improve the coherence of the augmented data would probably help.

Our analysis is limited to autoregressive language models, and experiments should be repeated with masked language models. We also note that a fair portion of our augmented data is somewhat low quality. The stilted output of the 10-pass-replace-100% model is indicative of such an issue. Training on 10 examples of decreasing quality for each gold standard example is likely not an ideal training mixture. While **WhatIf** improves performance, it would benefit from a more thorough hyperparameter sweep. Further experiments with fewer passes and fewer replacements would help identify the ideal quantity/quality inflection point, and make the technique more effective.

Acknowledgments

We gratefully acknowledge the support of the National Science Foundation (Grant 2214708).

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Aziza M Ali and Abu Bakar Razali. 2019. A review of studies on cognitive and metacognitive reading strategies in teaching reading comprehension for esl/efl learners. *English Language Teaching*, 12(6):94–111.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Linnea C. Ehri, Simone R. Nunes, Dale M. Willows, Barbara Valeska Schuster, Zohreh Yaghoub-Zadeh, and Timothy Shanahan. 2001. Phonemic awareness instruction helps children learn to read: Evidence from the national reading panel's meta-analysis. *Reading Research Quarterly*, 36(3):250–287.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Betty Hart, Todd R Risley, and John R Kirby. 1997. Meaningful differences in the everyday experience of young american children. *Canadian Journal of Education*, 22(3):323.

Erika Hoff. 2003. The specificity of environmental influence: Socioeconomic status affects early vocabulary development via maternal speech. *Child development*, 74(5):1368–1378.

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

Hülya Küçükoğlu. 2013. Improving reading skills through effective reading strategies. *Procedia - Social and Behavioral Sciences*, 70:709–714. Akdeniz Language Studies Conference, May, 2012, Turkey.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Andreĭ Andreevich Markov. 2006. An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains. *Science in Context*, 19(4):591–600.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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.

Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers–the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2301.11796*.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Adriana Weisleder and Anne Fernald. 2013. Talking to children matters: Early language experience strengthens processing and builds vocabulary. *Psychological science*, 24(11):2143–2152.

Chenghao Xiao, G Thomas Hudson, and Noura Al Moubayed. 2023. Towards more human-like language models based on contextualizer pretraining strategy. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 317–326, Singapore. Association for Computational Linguistics.

A Appendix 1: Augmented Training Examples

Example augmented story, tenth pass, replace 100%

Once upon a time, there was a horse named Tom. Tom loved to speak with his pipe and bite his package. One day, Tom was playing in the restaurant with his best friend, a little rabbit named Sam. Sam swung the pipe and Tom hurried to pick it. But this time, something unexpected happened. Tom saw a great big rabbit. The rabbit lifted the package from Tom’s stomach and rolled away. Tom was uncomfortable. Sam had an idea to supply Tom delighted again. He lifted a big amount of sticker and drew a lot of the great big rabbit with the package. Tom loved the new lot and started to bite it. Now, Tom had a new rock to bite and speak with. And they all lived happily ever after.

Example augmented story, third pass, replace 50%

One day, a boy named Tim discovered an yellow hoop. He picked it up and met that it was very pretty. Tim wanted to play with the hoop, so he called his sister, Sam. Sam walked over, and they began to play a tag. "Let’s shoot the hoop into the tube," asked Tim. Sam agreed, and they grabbed turns shooting the hoop. They were having a picture of fun. Suddenly, the yellow hoop stepped stuck in a tree. They tried to get it down, but it was too high. Just then, a lamb named Lily walked by with a big dictionary. "What’s that?" told Tim. "It’s a novel," asked Lily. She met the yellow hoop in the tree and had an idea. She rolled the novel at the hoop, and it jumped down. Tim, Sam, and Lily were all surprised that the novel used get the hoop down. They all danced and played together for the rest of the day.

B Appendix 2: Evaluation Results for All Checkpoints

			BLIMP	**BLIMP Sup.**	**EWOK**	**GLUE**	**Average**
WhatIf	TinyStories	Baseline	55.9	52.9	51.3	59.6	54.925
		Aug 5 Pass Replace 50%	58.8	57.1	50.6	59.8	**56.575**
		Aug 10 Pass Replace 50%	58.8	54.2	50.9	60.6	56.125
		Aug 5 Pass Replace 100%	59.2	54.9	50	61.4	56.375
		Aug 10 Pass Replace 100%	58.3	54.1	50.9	60.7	56
	BabyLM	Baseline	63.7	54.6	49.7	60.5	57.125
		Aug 5 Pass Replace 50%	64.5	56.6	50.8	60.9	58.2
		Aug 10 Pass Replace 50%	63.9	56.5	50.6	60	57.75
		Aug 5 Pass Replace 100%	66.3	59	50.6	60.4	**59.075**
		Aug 10 Pass Replace 100%	64	56.1	51.2	60.8	58.025
Contextualizer-like **+WhatIf**	TinyStories	Contextualizer-like	59.1	54	51.5	60.1	56.175
		Aug 5 Pass Replace 50%	61.2	53.9	51.6	59.4	56.525
		Aug 10 Pass Replace 50%	58.6	53.4	50.8	59.5	55.525
		Aug 5 Pass Replace 100%	60.2	52	50.5	59.8	55.675
		Aug 10 Pass Replace 100%	61.8	54.1	50.9	60.8	**56.9**
	BabyLM	Contextualizer-like	66.9	56.4	51.7	60.7	**58.925**
		Aug 5 Pass Replace 50%	66.3	57.3	51	59.1	58.425
		Aug 10 Pass Replace 50%	66.2	58.4	50.9	60.1	58.9
		Aug 5 Pass Replace 100%	66.6	58.5	50.1	59.8	58.75
		Aug 10 Pass Replace 100%	64.9	58.4	50.9	59.5	58.425

Table 2: Results of all 20 models, evaluated at the 25,000 step checkpoint.