Data Augmentation for Low-resourced Language Modeling

Stanford CS224N Custom Project

Wanyue Zhai
Department of Computer Science
Stanford University
wzhai702@stanford.edu

Shubo Yang
Department of Electrical Engineering
Stanford University
shuboy@stanford.edu

Abstract

The language models nowadays have seen an increasing number of tokens for training. However, many languages and systems cannot satisfy the high data demand, and it takes significant money and time to collect data. Thus, our project focuses on addressing low-resource language modeling using data augmentation. This is inspired by the BabyLM challenge, which trains language models based on data heard by 13-year-old children. We applied two compositional data augmentation approaches: GECA and resampling. The augmented data are carefully analyzed and evaluated by surprisal with respect to the base model. The models trained on the augmented data are evaluated by perplexity and BLiMP. The results show that resampling achieves better performance than GECA in terms of perplexity in our experimental setting. The perplexity performance improvement is related to the size of the augmented data. Additionally, the BLiMP accuracy for different augmented data sets shows little difference. Our experiment analysis and evaluation results can greatly help to understand the data augmentation strengths and drawbacks and help future model fine-tuning.

1 Key Information to include

- Mentor (custom project only): Jesse Mu, Shikhar Murty

2 Introduction

Natural Language Processing (NLP) systems have always struggled with the issue of limited data. Out of the 7,000 existing languages, only a few have adequate NLP-related resources. Even for NLP systems that build on top of those languages, most supervised systems require a large amount of manually annotated data in order to obtain higher performance, which takes significant time, money, or expertise for labels. Therefore, there is also a constant focus on achieving better performance on NLP systems with a limited amount of data. While supervised systems suffer from low-resource problems, language model training also suffers from a shortage of data. With the development of large language models in recent years, much attention has been focused on increasing the number of parameters and the size of datasets when training language models. Most of the well-known language models (BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), Chinchilla (Hoffmann et al., 2022)) are trained on billions and even trillions of data.

A common and traditional approach to tackle the problem of limited resources is data augmentation, which generates new data by transforming existing data points and based on prior knowledge about the problem's structure. However, unlike computer vision, where manipulation of images is easy, the discrete nature of textual data and complex syntactic structures make data augmentation in NLP often hard.

Recent research tackled the augmentation problem by integrating compositional features where different fragments from different sentences are re-combined to create augmented examples (Chen et al., 2023). These methods have more designed rules instead of random swapping but have higher compositional generalization abilities. There are two methods of compositional augmentation that we use in this paper: "Good-Enough Compositional Data Augmentation" (GECA) (Andreas, 2020) and resampling (He et al., 2019). The project is also inspired by the shared task, the BabyLM challenge (Warstadt et al., 2023). Given the data constraints similar to the learning for 13-year-old children (less than 10 million words), we train language models with limited resources using two data augmentation methods. [1] The contributions of this project can be summarized as follows:

- Applied two augmentation methods, GECA and resampling, to the BabyLM dataset and conducted qualitative checks with the augmented sentences.
- The performances are compared between adding more GECA augmented data, resampling augmented data, and adding extra data from the datasets. The augmented data is evaluated by surprisal, and the models are evaluated by perplexity and BLiMP tasks.
- The strengths and limitations of two data augmentation methods are carefully analyzed for future optimization of the language model pretraining.

3 Related Work

Data augmentation has been studied extensively in natural language processing. They generate new data based on the existing training sets. These methods can be split into token-level augmentation, sentence-level augmentation, and adversarial methods. Token-level augmentation modifies words and phrases in the sentence without breaking the original syntax and semantics. For example, Wei and Zou (2019) uses synonym replacement, random insertion, random swap, and random deletion for creating new data. A common method for sentence-level data augmentation is paraphrasing. Among those methods, Sennrich et al. (2015) uses round-trip translation so that the meaning of the text remains unchanged, but is expressed in more diverse word choices. Morris et al. (2020) presents a data augmentation framework by adding small perturbation to the input so that it produces the adversarial example specified by the constraints.

In this work, we focus on two of those methods. Specifically, the "Good-Enough Compositional Data Augmentation" approach presented by Andreas (2020) is a token-level data augmentation method that leverages the compositional structure of language to generate “good enough” augmented training examples. Resampling (He et al., 2019), on the other hand, uses the word distribution from the base model and generates new sentence-based data that aligns with the original training data.

4 Approach

4.1 GECA

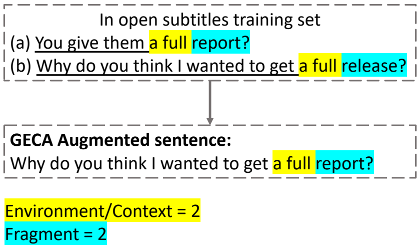
GECA identifies if two training samples appear in similar environments, then generate new data using the environments for the first fragment after substituting the second fragment. This method is based on the assumption that contexts provide great information and signals about a sentence fragment category.

An augmented example is shown in Figure 1a. Given a sequence of words, there are three key elements:

- Fragment: a set of non-overlapping spans of w, e.g., the blue-highlighted words.
- Template: the sequence of words w with fragments removed, e.g., ’You give them a full ...’ in (a)
- Environment: a template restricted to k-word window, e.g., $k = 2$, in (a) the fragment is ’report?’, the environment spans two words before, so the environment is ’a full’.

After identifying and matching the environments, new data can be generated by substituting ’release?’ with ’report?’. This leads to the GECA augmented sentence in Figure 1a.

[1] The code for model training and evaluations can be found on https://github.com/jayelm/stanford-babylm/tree/data_augmentation



In open subtitles training set

(a) You give them a full report?

(b) Why do you think I wanted to get a full release?

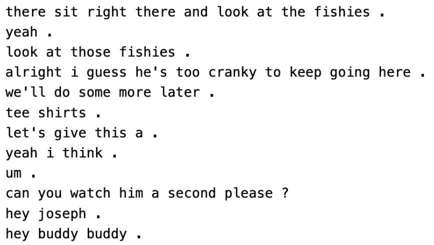
GECA Augmented sentence:

Why do you think I wanted to get a full report?

Environment/Context = 2

Fragment = 2

(a) Example of a GECA augmented sentence



```
there sit right there and look at the fishies .
yeah .
look at those fishies .
alright i guess he's too cranky to keep going here .
we'll do some more later .
tee shirts .
let's give this a .
yeah i think .
um .
can you watch him a second please ?
hey joseph .
hey buddy buddy .
```

(b) Example of Text generated from resampling

4.2 Resampling

The resampling data augmentation approach is similar to self-training (He et al., 2019; Akyürek et al., 2020) as seen in most semi-supervised methods. It takes a base model trained on existing training data (commonly labeled data) so that it can label the unlabeled data, thus creating new guided labeled data for further training. The setting in this paper is different from the previous self-training in that we do not use and produce labeled data. Instead, we feed real child-directed texts to the base model and use the learned distribution from the base model to generate new child-directed data. Figure 1b shows examples of generated text.

After we get the generated data from the two data augmentation methods, we train the language model with the same architecture as the one without any data augmentation. To compare the performance of data augmentation with simply adding more data, we extract the same amount of data from each of the remaining dataset domains. We report the performance of each model trained with 1) different numbers of generated data and 2) different data augmentation methods.

5 Experiments

5.1 Data

We use the dataset provided by the BabyLM challenge (Warstadt et al., 2023), whose task is to mimic the learning of 13-year-old children. Therefore, the datasets are selected based on the common inputs to children. The data are sampled from 10 different domains that are common as inputs to children. We use 10M of the data for our training of the baseline models. The details are shown in Table 1. The models are evaluated on a different cluster of the same datasets and contain 9.4M words in total.

Dataset	Domain	# words
CHILDES (MacWhinney, 2000)	Child-directed speech	0.44M
British National Corpus (BNC), dialogue portion [2]	Dialogue	0.86M
Children's Book Test (Hill et al., 2016)	Children's books	0.57M
Children's Stories Text Corpus [3]	Children's books	0.34M
Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2020)	Written English	0.99M
OpenSubtitles (Lison and Tiedemann, 2016)	Movie subtitles	3.09M
QCRI Educational Domain Corpus (QED) (Abdelali et al., 2014)	Educational video subtitles	1.04M
Wikipedia [4]	Wikipedia (English)	0.99M
Simple Wikipedia [5]	Wikipedia (Simple English)	1.52M
Switchboard Dialog Act Corpus (Stolcke et al., 2000)	Dialogue	0.12M
Total		9.96M

Table 1: The datasets released for the STRICT-SMALL tracks of the BabyLM Challenge

5.2 Evaluation Methods

We evaluate our model performance under a fixed computation budget in two ways: perplexity and BLiMP Warstadt et al. (2020). Perplexity is one of the most commonly used evaluation metrics for the evaluation of language models. It is measured as shown in Equation 1

$$PPL(X) = exp(-\frac{1}{t}\sum_{i}^{t} log_{p\theta}(x_i|x_{<i})) \ , \tag{1}$$

where $X = (x_0, x_1, ..., x_t)$ is a tokenized sequence and $log_{p\theta}(x_i|x_{<i})$ is the log-likelihood of the ith token conditioned on all the preceding tokens.

We also evaluate the model on its ability to make targeted syntactic judgments. To do so, we use the BLiMP dataset Warstadt et al. (2020) and the software lm-evaluation-harness Gao et al. (2021). BLiMP is a dataset consisting of 67 individual datasets, each with 1,000 minimal pairs that differ grammatically, but have the same meaning. The grammatical differences are separated into 12 distinct linguistic phenomena (e.g. agreement, control/raising, wh-extraction) for evaluation. An example of a minimal pair that differs in binding is:

- Acceptable example: Carl said Lori helped him.
- Unacceptable example: Carl said Lori helped himself.

The task of BLiMP is a forced-choice task that evaluates the model's ability to distinguish the bad sentence from the good sentence in all minimal pairs. If the score assigned to the good sentence is higher than the bad sentence, we consider it a correct assignment. The results of the task will be shown as both overall accuracy and distinct linguistic phenomena accuracy.

5.3 Experimental Details

5.3.1 Baseline

We first explored the best model architecture to perform the task. Hence, we do not use any data augmentation measures in the baseline model. To find the best model architecture, we ran 3 experiments. They are 6-layered GPT-2 with 6 attention heads, 6-layered GPT-2 with 12 attention heads, and 12-layered GPT-2 with 12 attention heads respectively. The other configurations are kept consistent. More specifically, we use a learning rate of 5.0e-05, linear schedular, adam optimizer, trained through 11 epochs. The training for all 6-layered models takes around 4.5 hours, and the 12-layered model takes around 7.5 hours.

5.3.2 Experiments

GECA data augmentation We use GECA to first generate augmented data(Andreas, 2020), with code [6]. For every category of the data, we feed the files into GECA code, and adjust the parameters to generate data with different surprisal listed in 5.4.2. The general setting is environment token length $context = 2$, length of fragment $wug\_size = 2$, number of fragment $wug\_count = 1$, the number of different fragments to swap in the environment $variant = 1, 3, 5$, the number of generated samples $n\_sample = 1000$, maximum sequence length to compare for finding the environment $max\_comp\_len = 100, 300, 500, 800$. Note that for some specific files, there are not many matching environments, so the generated data size may be a little smaller.

Resampling data augmentation For data augmentation through resampling, we use the model that is trained on the original BabyLM dataset (as shown in Table 1 to generate more sentences. We use Top-p sampling for text generation as mentioned in von Platen (2020). Top-p sampling chooses the next words randomly from the smallest set of words that exceeds the probability threshold p, so the size of word choices dynamically changes according to the probability distribution. To avoid duplicated sentences, we extract the first word of each sentence from the original BabyLM dataset and generate sentences that have a maximum length equal to the original sentence. The time required

[6] The code for GECA data augmentation can be found on https://github.com/yangs12/GECA-modified.git

to generate the sentences increases with the number of words and can take more than 12 hours to generate 2 million additional words. Due to time and computation constraints, we generated datasets containing 232k, 447k, 971k, and 2M additional words.

Model Training We train the language models using the previous most efficient architecture: a 6-layered model with 12 attention heads. We extract the same amount of data from each remaining dataset domain to maintain the same word increase. Each model is run using the data from the original dataset as described in Table 1 along with either generated data or extracted real data. Table 6 in Appendix A.1 shows an example of a comparison of word increase when generating 1k sentences in GECA data augmentation, and the corresponding increase in real and resampling data. It is also worth noting that all the experiments are compared under fixed computation budgets (11 epochs), so the experiments done with the same increase in data size are evaluated after the same amount of steps. As we stop at the maximum epoch 11, the training time increases as the training dataset size increases. It takes around 5 hours to train the model with additional 232k words, and more than 10 hours to train with 5M additional words.

5.4 Results

5.4.1 Baseline

Table 2 shows the perplexity evaluated on the three baseline models. It can be observed that the perplexity is very similar among the different architectures. We compared our model of 6-layered GPT-2 with 12 attention heads and the model of 12-layered GPT-2 with the original GPT-2 model. As we can see, in general, the two custom models have very little difference in performance. Therefore, we chose the most efficient architecture (6-layered model with 12 attention heads) to proceed to train with data augmentation.

	6 layers	12 layers
6 attention heads	49.566	-
12 attention heads	51.971	48.092

Table 2: Perplexity on Baseline Models

5.4.2 Evaluation using Surprisal

GECA augmented data are evaluated by surprisal, which is the perplexity with respect to the base model trained on existing data in BabyLM dataset. This reflects how surprisingly the GECA data compared with our training data, for further augmentation evaluation. The main hyperparameters that we adjust are $variants$ and $max\_compare\_length$. As seen in Table 3, more variants and longer compare length lead to lower surprisal. This is because more fragments swapped into one environment and more environment comparing length result in more similar environments in our augmented data. The data with more similar environments are less surprising than diverse environments.

GECA Params

context	variants	max compare length	Surprisal
2	1	**100**	66.5911
2	1	**300**	63.6386
2	1	**500**	61.0856
2	1	**800**	61.0493
2	**3**	300	59.2269
2	**5**	300	57.0991

Table 3: Surprisal on GECA data

5.4.3 Evaluation using Perplexity

Figure 2 shows our evaluation score for perplexity when training on baseline (without additional data), original dataset with additional GECA data, original dataset with additional resampling data, and original dataset with additional real data. The experiments with the same increase in data size are evaluated after the same amount of steps.

It is expected and can be observed that adding real data always helps reduce the perplexity score. We can also see that GECA data reduce perplexity when the dataset size is relatively small. But the perplexity score increases when the GECA data size is too large. It is expected that resampling would help reduce perplexity, but quite surprising that it can achieve a perplexity close to adding real data when the data increase is small. Based on perplexity evaluation, we know that both GECA and resampling can help reduce perplexity when the data size increase is small, and that resampling is generally a more effective data augmentation strategy under the given setting.

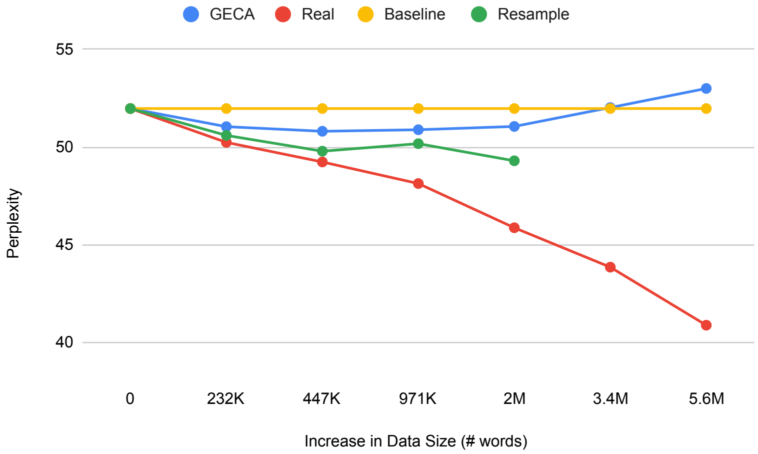


Figure 2: Results on Perplexity across different data size

5.4.4 Evaluation using BLiMP tasks

The results shown in Figure 3 show our average BLiMP task accuracy when trained on baseline (without additional data), original dataset with additional GECA data, original dataset with additional resampling data, and original dataset with additional real data. A more detailed decomposition that shows the accuracy of each BLiMP linguistic phenomenon is shown in Figure 5 in the Appendix. The score shown here is an average of all BLiMP tasks. It can be observed that the average accuracy for BLiMP tasks does not show as much difference as shown in perplexity. The BLiMP average when tested on resampling almost overlapped with the baseline model. The BLiMP accuracy for GECA decreased at first, but increased for 2M data increase, but never reached higher than the baseline. The real performance first decreases below the baseline, and later increases after adding 2M real data. All changes in accuracy are within 5% and are quite far below the standard GPT-2 performance.

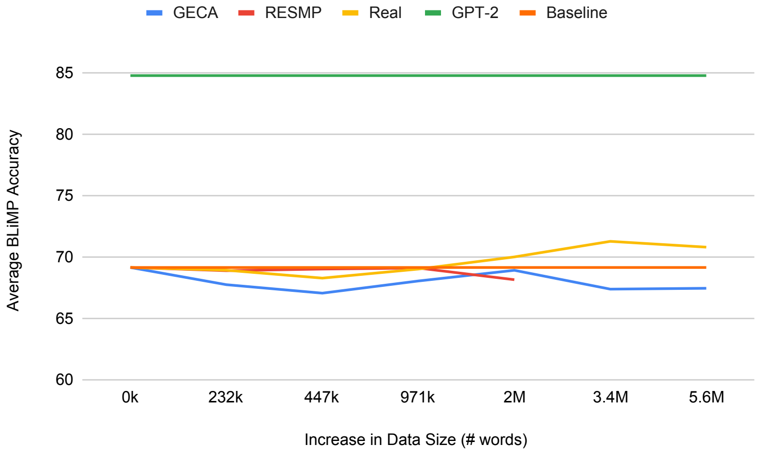


Figure 3: Results on BLiMP accuracy across different data size

5.4.5 Relationship between Perplexity and Surprisal

The relationship between perplexity and surprisal is plotted in Figure 4. The resampling augmented data has lower surprisal and the GECA augmented data always have higher surprisal. The relationship is expected to be ’U’-shape, where too low surprisal cannot let the model learn new things, and meanwhile too high surprisal may not be reasonable data. Thus, the point of real data may be the optimal point with the surprisal and perplexity curve. In future work, we will run more experiments with data augmentation and try to achieve the performance as real data.

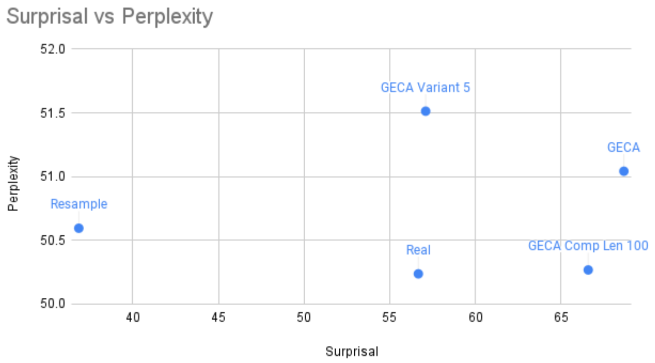


Figure 4: Results on surprisal and perplexity relationships.

6 Analysis

6.1 Augmented Data

We now look more closely into the augmented data that is produced by each method to explore the performance difference.

Resampling Since resampling generates new sentences based on previous distribution, we calculated the amount of generated sentences that have an exact match in the original dataset. It is shown that around 15.8% - 17.65% of the resampling dataset is repeating the data in the original dataset. Most of the duplicated sentences are short sequences of words as shown in 4. Despite the duplicates, the new model is able to receive data that is not seen before and learn based on the previous distribution.

Seen sentences	Unseen sentences
I love you	I just think we should have a team
Just a minute.	Just don’t tell me?
That’s enough.	That’s where it’s been on the car yet
You go.	You are like my mom.

Table 4: Example Generated Data from Resampling in OpenSubtitles

GECA GECA will not have the same sentences as original data, because the sentences are generated by replacing fragments. However, this means the augmented data will have a large amount of overlap with the original sentence. While GECA can generate some sentences retaining the compositional structures, it also can generate unreasonable sentences. The unreasonable sentences are expected and acceptable, since fully integrating grammar and syntax needs manual labelling, and what GECA does is to try best for compositional data augmentation. The augmented examples from GECA are listed in Table 5. These examples are taken from the augmented data generated based on "open subtitles", and other examples are shown in Appendix A.3. It can also be seen that most fragments are short and at the end of a sentence. This is because the environment context of the ending fragment only need to match the forehand words, which makes it easier to match.

Original Environment	Fragment	Augmented Sentence
Do we make sense as a couple?	Butterfly?	Do we make sense as a Butterfly?
Sorry, I continued to believe you were Teresa.	Leo.	Sorry, I continued to believe you were Leo.
I did it while you were unconscious.	bored.	I did it while you were bored.
Yeah, there was a retard	Five!	Yeah, there was a Five!
Well, it looks like it was a temple.	Hey,Bella.	Well, it looks like it was a Hey,Bella.

Table 5: GECA generated data examples.

6.2 Analysis on Evaluation Results

The evaluation results on Figure 2 show that the performance of language modeling does not simply increase as the amount of data they receive increases. In addition, we know that GECA data helps the model when the data added to training is a small portion of the training samples (2%-20%), but hurts the model performance when it makes up a large portion of the training samples (35%-50%). Resampling always increases model performance and can achieve a score close to adding real data when the added data is less than 5%. The differences between resampling and real data increase as the data size increases.

While the result shows the trend, since the current result is tested with the same number of epochs, we can only conclude our results based on the setting. As resampling uses the learned distribution from the base model instead of the real text distribution of words, we do not expect the increase in performance in resampling to persist when the computation resource is unlimited. It is also worth pointing out that although the final model is trained with the same number of epochs, resampling requires extra training for the base model in order to generate new samples, but GECA only needs to train a single model.

While the results on perplexity show a clear trend between performance and dataset increase, the evaluation using BLiMP is less indicative. When we look more closely at the detailed performance of each model on each linguistic phenomenon (Figure 5), we can see that there is no consistent trend in the best-performing model. The best model for evaluation accuracy for each task is different. The differences in average BLiMP accuracy are small (within 5%) for each experiment. Observing from the trend in Figure 3 when adding more GECA or resampled data, we know that adding augmented data does not improve the overall BLiMP evaluation results. With the limited amount of increase in real data, the difference in BLiMP accuracy is also small compared to the baseline model. Therefore, the limited amount of data increase as we tested in this paper is not enough for a large BLiMP performance difference.

7 Conclusion and Future Work

To tackle the challenge of limited-resource language modelling, we applied two data augmentation methods on BabyLM dataset: GECA and resampling. The augmented data are analyzed and the performance is evaluated by surprisal, perplexity, and BLiMP accuracy. Both methods show performance improvement on perplexity, and resampling demonstrates better result on our dataset. The experiments on BLiMP performance does not have significant differences. The analysis of these evaluation results can help us compare the strengths of the augmentation methods, understand the schemes of data augmentation, and further finetune the models.

For future work, first, we will run new experiments with the criteria of minimum achievable loss, compared to the limited computation resource in this paper. Second, more experiments will be conducted on perplexity and data size, and perplexity and surprisal, where we expect a 'U'-shaped curve. Third, another line of comparison of adding the seen data of the base model will be established, in addition to adding unseen real data for now. Finally, we will explore other possible data augmentation methods, and compare on their effectiveness on this task.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In *LREC*, volume 14, pages 1044–1054.

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2020. Learning to recombine and resample data for compositional generalization. *arXiv preprint arXiv:2010.03706*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. 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, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. arxiv 2015. *arXiv preprint arXiv:1511.02301*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

Patrick von Platen. 2020. How to generate text: using different decoding methods for language generation with transformers.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

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.

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 and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

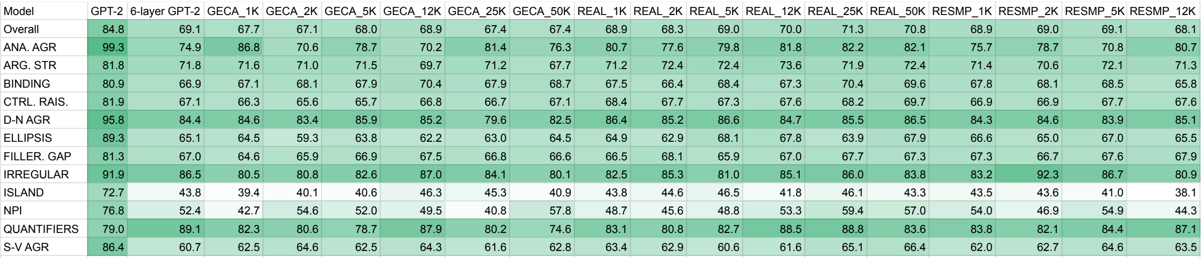
A Appendix

A.1 Dataset breakdown for real data and GECA data

Dataset	GECA	REAL	Resamp
CHILDES (MacWhinney, 2000)	8,139	8,142	8,144
British National Corpus (BNC), dialogue portion [7]	18,417	18,423	18,423
Children's Book Test (Hill et al., 2016)	27,616	27,642	27,645
Children's Stories Text Corpus [8]	18,807	18,820	12,146
Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2020)	11,058	11,059	11,062
OpenSubtitles (Lison and Tiedemann, 2016)	9,884	9,893	9,898
QCRI Educational Domain Corpus (QED) (Abdelali et al., 2014)	13,846	13,868	13,868
Wikipedia [9]	54,679	54,725	54,725
Simple Wikipedia [10]	36,050	36,076	36,083
Switchboard Dialog Act Corpus (Stolcke et al., 2000)	34,014	34,019	34,021
Total	232,510	232,667	226,015

Table 6: Increase in number of words for GECA vs Real Data

A.2 Detailed BLiMP Evaluation



Model	GPT-2	6-layer GPT-2	GECA_1K	GECA_2K	GECA_5K	GECA_12K	GECA_25K	GECA_50K	REAL_1K	REAL_2K	REAL_5K	REAL_12K	REAL_25K	REAL_50K	RESMP_1K	RESMP_2K	RESMP_5K	RESMP_12K
Overall	84.8	69.1	67.7	67.1	68.0	68.9	67.4	67.4	68.9	68.3	69.0	70.0	71.3	70.8	68.9	69.0	69.1	68.1
ANA. AGR	99.3	74.9	86.8	70.6	78.7	70.2	81.4	76.3	80.7	77.6	79.8	81.8	82.2	82.1	75.7	78.7	70.8	80.7
ARG. STR	81.8	71.8	71.6	71.0	71.5	69.7	71.2	67.7	71.2	72.4	72.4	73.6	71.9	72.4	71.4	70.6	72.1	71.3
BINDING	80.9	66.9	67.1	68.1	67.9	70.4	67.9	68.7	67.5	66.4	68.4	67.3	70.4	69.6	67.8	68.1	68.5	65.8
CTRL. RAIS.	81.9	67.1	66.3	65.6	65.7	66.8	66.7	67.1	68.4	67.7	67.3	67.6	68.2	69.7	66.9	66.9	67.7	67.6
D-N AGR	95.8	84.4	84.6	83.4	85.9	85.2	79.6	82.5	86.4	85.2	86.6	84.7	85.5	86.5	84.3	84.6	83.9	85.1
ELLIPSIS	89.3	65.1	64.5	59.3	63.8	62.2	63.0	64.5	64.9	62.9	68.1	67.8	63.9	67.9	66.6	65.0	67.0	65.5
FILLER. GAP	81.3	67.0	64.6	65.9	66.9	67.5	66.8	66.6	66.5	68.1	65.9	67.0	67.7	67.3	67.3	66.7	67.6	67.9
IRREGULAR	91.9	86.5	80.5	80.8	82.6	87.0	84.1	80.1	82.5	85.3	81.0	85.1	86.0	83.8	83.2	92.3	86.7	80.9
ISLAND	72.7	43.8	39.4	40.1	40.6	46.3	45.3	40.9	43.8	44.6	46.5	41.8	46.1	43.3	43.5	43.6	41.0	38.1
NPI	76.8	52.4	42.7	54.6	52.0	49.5	40.8	57.8	48.7	45.6	48.8	53.3	59.4	57.0	54.0	46.9	54.9	44.3
QUANTIFIERS	79.0	89.1	82.3	80.6	78.7	87.9	80.2	74.6	83.1	80.8	82.7	88.5	88.8	83.6	83.8	82.1	84.4	87.1
S-V AGR	86.4	60.7	62.5	64.6	62.5	64.3	61.6	62.8	63.4	62.9	60.6	61.6	65.1	66.4	62.0	62.7	64.6	63.5

Figure 5: Detailed Accuracy on BLiMP tasks

A.3 GECA Augmented sentences

1. From "QED"
   Environment: BUT I LOVE YOU.
   Fragment: IT!
   Augmented data: BUT I LOVE IT!

2. From "QED"
   Environment: We're going to ride in the elevator.
   Fragment: Standby.
   Augmented data: We're going to ride in the Standby.

3. From "gutenberg" Environment: peeping in the shops, we see one and go in and examine it. It is quite a
   Fragment: Climate
   Augmented data: Climate in the shops, we see one and go in and examine it. It is quite a

4. From "wikipedia"
   Environment: On August 19, 2009, for unspecified personal reasons, Maurizio Bianchi decided again to completely stop making music.
   Fragment: Grotesque, (chess)
   Augmented data: On August 19, 2009, for unspecified personal reasons, Grotesque (chess) decided again to completely stop making music. This decision was soon after reversed; Grotesque (chess) continued to release new music.

5. From "Children stories"
   Environment: "Well! be off with you," said the Troll.
   Fragment: rabbit.
   Augmented data: "Well! be off with you," said the rabbit.
6. From "BNC spoken"
   Environment: Perhaps he's flying down the chimney.
   Fragment: Terry's
   Augmented data: Perhaps he's flying down the Terry's