Baby's CoThought: Leveraging Large Language Models for Enhanced Reasoning in Compact Models

Zheyu Zhang[*♣] **Han Yang**[*♣,♢] **Bolei Ma**[*♠] **David Rügamer**[♠,♡] **Ercong Nie**[†♣,♡]

[♣]Center for Information and Language Processing, LMU Munich
[♢]GESIS - Leibniz Institute for the Social Sciences, Cologne
[♠]Department of Statistics, LMU Munich [♡]Munich Center for Machine Learning

zheyu.zhang@campus.lmu.de han.yang@gesis.org
{bolei.ma, david.ruegamer}@stat.uni-muenchen.de
nie@cis.lmu.de

Abstract

Large Language Models (LLMs) demonstrate remarkable performance on a variety of natural language understanding (NLU) tasks, primarily due to their in-context learning ability. This ability could be applied to building baby-like models, i.e. models at small scales, improving training efficiency. In this paper, we propose a "**CoThought**" pipeline, which efficiently trains smaller "baby" language models (BabyLMs) by leveraging the **C**hain **o**f **Thought** prompting of LLMs. Our pipeline restructures a dataset of less than 100M in size using GPT-3.5-turbo, transforming it into task-oriented, human-readable texts that are comparable to the school texts for language learners. The BabyLM is then pretrained on this restructured dataset in a RoBERTa fashion. In evaluations across 4 benchmarks, our BabyLM outperforms the vanilla RoBERTa in 10 linguistic, NLU, and question-answering tasks by more than 3 points, showing a superior ability to extract contextual information. These results suggest that compact LMs pretrained on small, LLM-restructured data can better understand tasks and achieve improved performance.[1]

1 Introduction

Recent advances in language modeling of Large Language Models (LLMs) have shown great performance potential on diverse NLP tasks. A large number of work has been proposed towards enhancing LLMs pretraining at massive scales (Devlin et al., 2019; Radford and Narasimhan, 2018; Brown et al., 2020). However, less attention has been paid to language model (LM) pretraining at smaller human-like data scales, i.e. smaller data scales, which are similar to the amount of language data for human language acquisition.

Studies in language acquisition demonstrate that humans predominantly acquire language in early life stages by observing their environment. Significant progress in language communication and usage is typically achieved by early childhood (Tomasello, 2003; Saxton, 2010). Previous studies show that language modeling is to some extent similar to children's language acquisition, as they both require input data from the outside world and learn the data by updating knowledge about the outside world repeatedly (Nikolaus and Fourtassi, 2021; Chang and Bergen, 2022; Evanson et al., 2023). It is reasonable to apply this human cognitive process to LM pretraining by using relatively small sets of pretraining data that are comparable to the text data for human language acquisition.

While a child learns a piece of knowledge by continuously obtaining relevant examples from the outside world and updating its knowledge base, pretrained LLMs have the capacity to learn and complete previously unknown tasks when given several task samples or instructions already from the inside of their context, the process of which is known as "*In-Context Learning*" (ICL) (Brown et al., 2020). A more recent advance of ICL called "*Chain of Thought*" (CoT) (Wei et al., 2022) significantly enhances the reasoning abilities of LLMs. CoT enables LLMs to perform a series of intermediate reasoning steps by providing a few CoT demonstrations as examples during the training process. This method has been found to be very effective, especially in complex reasoning tasks.

The LLM is like a teacher who is able to transfer knowledge by reformulating raw data from the outside world into a task-like text format by CoT prompting, making the data more suitable for teaching. The BabyLM is like a student who is trained

[*] Equal contribution.
[†] Corresponding author.
[1]The code for data processing and model training is available at: https://github.com/oooranz/Baby-CoThought.

Proceedings of the 27th Conference on Computational Natural Language Learning: Volume 2: The BabyLM Challenge, pages 158–170

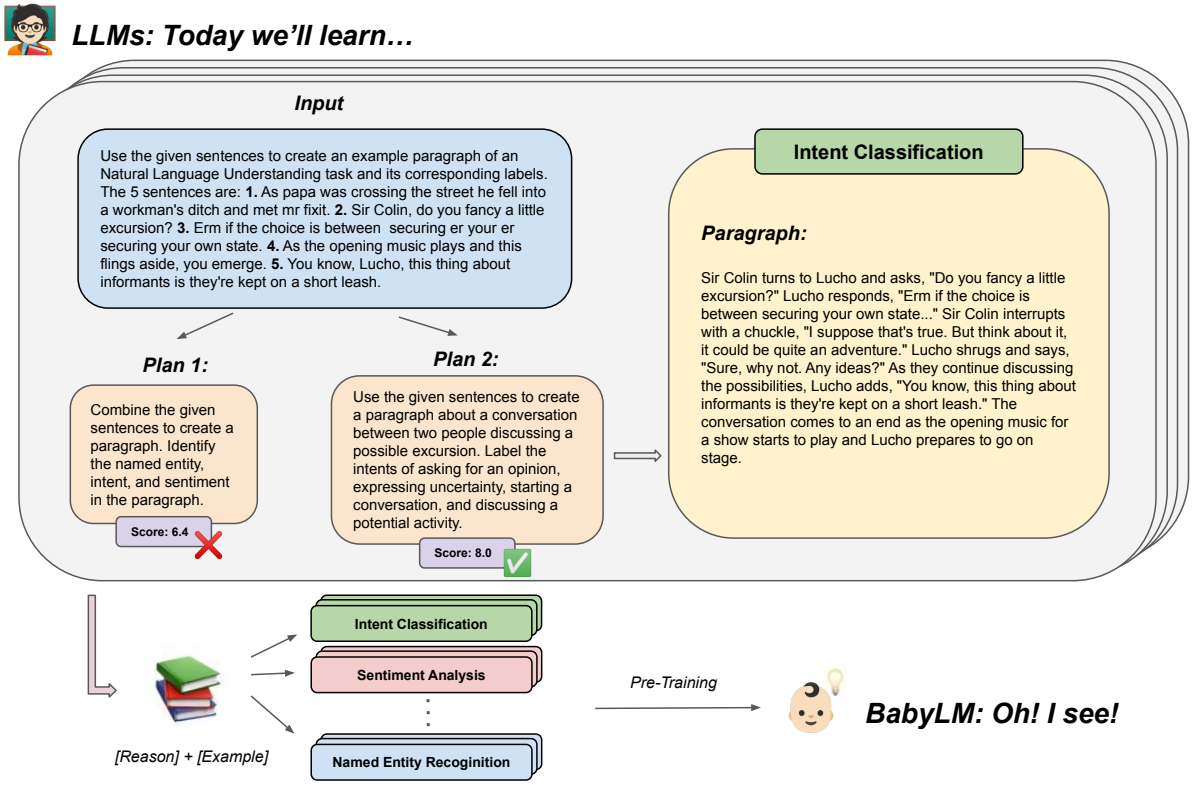

Figure 1: Overview of the "CoThought" pipeline. We propose to generate NLU examples from discrete short sentences using CoT prompting and an automatic scoring mechanism. This constructs a pretraining dataset in a *[Reason] + [Example]* format, which is then used to pretrain smaller models.

based on this generated text. In this work, we propose "**CoThought**" pipeline to pretrain a BabyLM with human-like smaller corpus data, by leveraging the LLM's **C**hain **o**f **Thought** feature and the child's cognitive learning ability. In this way, the LLM and the child are "co-thinking" during the training process. We use the "CoThought" approach to train our BabyLM, combining the productivity of the LLM with the effectiveness of human language acquisition for LM pretraining.

Our overall framework is illustrated in Figure 1. The raw pretraining data is provided by Warstadt et al. (2023) in the BabyLM Challenge, which has the goal of sample-efficient pretraining on a developmentally plausible corpus at a small human-like data scale. We choose the loose track of the BabyLM Challenge, where we apply our "CoThought" pipeline and use the LLM GPT-3.5-turbo[2] to preprocess the raw data. For every 5 sentences of the raw data, the GPT-3.5-turbo uses CoT prompting to propose different NLU tasks and selects the best task. Then, it combines these 5 sentences into a task-like text based on the best task for our BabyLM to learn. The BabyLM is pretrained on the augmented data in a RoBERTa (Liu et al., 2019) fashion. Our BabyLM pretrained in the CoThought pipeline notably outperforms the original RoBERTa model on common benchmarks.

Our work makes contributions in

1) proposing the CoThought pretraining pipeline fitting the human-like data scenarios,

2) pretraining a BabyLM model of the RoBERTa-base architect in the CoThought pipeline surpassing the original RoBERTa model on several tasks, and

3) providing insights of the CoThought pipeline by conducting linguistic case analysis on representative tasks.

2 Related Work

Language Acquisition and Modelling The language acquisition of children is a widely studied topic in linguistics. The empiricism of language acquisition contends that language ability is a component of social cognitive ability and children acquire

[2] https://platform.openai.com/docs/models/gpt-3-5

language through language communication and language use (Bybee, 2001; Pullum and Scholz, 2002; Tomasello, 2003; Saxton, 2010). According to the Universal Grammar (Chomsky, 1957), language norms and parameters are hard-wired within every single person, and learning a language is just a matter of adjusting those parameters (Gegov et al., 2014). In this way, child language acquisition and language modeling are similar, as the neural language models such as BERT (Devlin et al., 2019) and GPT (Radford and Narasimhan, 2018) are pretrained based on big corpora with their model parameters tuned during pretaining. Recent studies show the applicability of language models to child language development tracking. Nikolaus and Fourtassi (2021) propose an integrated perception- and production-based learning and highlight that children are not only understood as passively absorbing the input but also as actively participating in the construction of their linguistic knowledge in language learning. Chang and Bergen (2022) study the factors that predict words' ages of acquisition in contemporary language models compared to word acquisition in children. Evanson et al. (2023) compare the sequence of learning stages of language models with child language acquisition.

In-Context Learning (ICL) LLMs like GPT-3 (Brown et al., 2020) make "*In-Context Learning*" possible, which means the model makes predictions by learning from a natural language prompt describing the language task or learning from (only a few) examples. Based on the concept of ICL, recent research has demonstrated that LLMs can be used to extract relevant knowledge from the content. Liu et al. (2022) propose to use GPT-3 to generate pertinent contexts and then supply those contexts as extra input in order to answer a commonsense question. Yu et al. (2023) employ a generate-then-read pipeline which first prompts a large language model to generate contextual documents based on a given question, and then reads the generated documents to produce the final answer.

Chain of Thought (CoT) Wei et al. (2022) introduced "*Chain of Thought*", which is a series of intermediate reasoning steps a few chain of thought demonstrations are provided as exemplars in prompting, in order to improve the ICL ability of LLMs to perform complex reasoning. Kojima et al. (2023) demonstrate the zero-shot performance of CoT. Paranjape et al. (2023) introduces a framework that uses frozen LLMs to automatically generate intermediate reasoning steps as a program. Yao et al. (2023) put forward a "*Tree of Thoughts*" (ToT) framework, which generalizes over CoT to prompting language models and enables exploration over coherent units of text ("thoughts") that serve as intermediate steps toward problem solving. A more recent study (Gu et al., 2023) proposes a pretraining for ICL framework which pretrains the model on a set of "intrinsic tasks" in the general plain-text corpus using the simple language modeling objective to enhance the language models' ICL ability.

3 Method

In the realm of cognitive learning, the teacher's thought process greatly influences the way instructional content is delivered, which in turn impacts the students' understanding (Chew and Cerbin, 2021). Our method attempts to mimic this process. The LLMs, in the role of the teacher, use CoT prompting to reinterpret the raw data, generating task-like text that incorporates the context of the sentences and enriches the learning materials.

We first introduce an overview of our CoThought pipeline (see Figure 1 for an illustration) and then describe the details in the following sections.

3.1 Problem Statement

The genesis of our research lies in addressing a significant problem within the context of the BabyLM Challenge as proposed by Warstadt et al. (2023). The goal of this challenge is to conduct sample-efficient pretraining on a developmentally plausible corpus at a small human-like data scale, which we previously introduced. Nevertheless, the majority of the training data provided consists of discrete short sentences. As an illustration, below are some of the provided sentences:

```
- You want your book back, don't you?

- Let's see, do you want to see who this is?

- This is Big Bird.

- Enough with that.

- Can you read your book again? You like the
  book?
```

These sentences, albeit contextually rich, are sampled from a wide range of sources including dialogues, scripted content, fiction, nonfiction, and child-directed materials. Due to the diverse and fragmented nature of this dataset, the sentences

often lack strong semantic ties with each other, making it difficult for models to learn contextual and coherent representations.

In response, we propose a method that transforms these fragmented sentences into cohesive units using LLMs, subsequently enabling more effective learning for the smaller models. The succeeding sections will provide a succinct outline of our pipeline and process.

3.2 Creative NLU-Example Generation

Inspired by recent studies that demonstrate the capability of LLMs to generate rationales supporting their predictions, we invent a novel task called Creative NLU-Example Generation (CNLU-EG), inspired by the Creative Writing task proposed by the "*Tree of Thought*" (Yao et al., 2023). Instead of creating coherent paragraphs from random sentences, CNLU-EG employs the provided sentences to generate coherent paragraphs, which define a plausible intrinsic NLU task and its corresponding labels. In this task, we employ the reasoning capability of LLMs to generate rationales for training smaller baby models.

We first remove any duplicate sentences from the BabyLM_100M (Warstadt et al., 2023) D. After the cleaning process, we randomly sample five unique sentences $\{x_i\}_{i \in D}$ from the cleaned dataset D. We initiate the task by providing a specific CoT prompt p to the LLM. This prompt instructs the LLM to first create a plan, then use the provided sentences to compose an example paragraph that illustrates a possible intrinsic NLU task, and finally generate the corresponding labels for this task. Given the creative nature of the task, we use a zero-shot prompt here. The prompt is structured such that it encourages the LLM to present the output in four distinct sections: the plan, the paragraph, the task, and the labels.

Once the LLM receives the prompt p, for each sentence $x_i, i \in D$, the LLM generates an execution plan $\hat{r}_i$, a paragraph $\hat{e}_i$ embodying an example of a possible NLU task, the task name $\hat{t}_i$, and the corresponding labels $\hat{y}_i$.

CNLU-EG essentially transforms the original, discrete sentences into a structured task, anchoring the sentences to a common theme or question. This 'taskification' process helps to create a more cohesive narrative, enabling the baby model to gain a more contextual and comprehensive understanding of the sentences.

We also incorporate a scoring mechanism, to assess the coherence of the generated content. We use a separate simple zero-shot prompt, p_s, to instruct the LLM to analyze the composed paragraph and assign a coherence score ranging from 1 to 10. For each task output, the LLM generates five such coherence scores from the same scoring prompt p_s, and these scores are then averaged to produce a final coherence score. According to our settings, we explicitly direct the LLM to generate two distinct plans for each task. Each plan is independently scored, and the one that achieves a higher coherence score is selected for subsequent steps.

In this way, the LLM functions as a teacher, generating examples of possible NLU tasks, providing insights into how these examples were created, and supplying the corresponding labels. This collection of generated plans and example paragraphs forms the training data for the smaller model to learn from.

3.3 Training Data Construction

Our objective is to construct a high-quality dataset for pretraining our small model, ensuring the instances included in the training set are coherent and task-relevant. As previously discussed, each instance in our data comprises a tuple: an example e and a corresponding plan r, denoted as $[e, r]$. However, not all generated instances meet the quality criteria necessary for effective learning.

To filter out lower-quality instances, we employ the coherency score obtained through the p_s prompt. We set a threshold, stipulating that only instances with a coherency score of $s \geq 7.0$ are included in the training data. This threshold was empirically established based on extensive manual analysis to ensure a satisfactory level of coherence and quality in the dataset. Mathematically, this can be represented as:

$$D_{select} = [e_i, r_i] : i \in D, s_i \geq 7.0 \qquad (1)$$

Here, D denotes the initial set of generated instances and D_{select} represents the selected high-quality instances that are used for training.

Another important aspect of our methodology is leveraging the correlation between segments with similar intrinsic tasks. Studies indicate that such segments when grouped together, provide valuable information for ICL (Gu et al., 2023). Therefore, we aim to collate instances with similar tasks, denoted as T, into grouped sets, which we denote as

G_T.

$$G_T = [e_i, r_i] : i \in D_{select}, t_i = T \quad (2)$$

In the equation above, t_i represents the task type of the i-th instance, and G_T denotes the set of instances from D_{select} that are associated with task type T.

In the end, we amalgamate these grouped sets to create a comprehensive pretraining dataset containing N instances.

$$D_{pretrain} = \bigcup_{T \in \mathcal{T}} G_T \quad (3)$$

Here, $\mathcal{T}$ represents the set of all task types and G_T denotes the set of instances corresponding to each task type T in D_{select}.

Through these rigorous steps, we ensure that the final training data is both high-quality and task-relevant, optimally structured to facilitate effective learning in our small model.

4 Experimental Setups

We conducted our experiments in three parts, the generation of the additional data used for training, the pretraining of the language model, and the evaluation.

4.1 Data Generation via CoT Prompting

We generated first our extended data based on the dataset `babylm_100M` (Warstadt et al., 2023), which contains subsets including AOCHILDES, BNC spoken, cbt, children stories, Gutenberg, pen subtitles, qed, simple Wikipedia, switchboard, and Wikipedia.[3]

We leveraged the API of GPT-3.5-turbo from OpenAI and provided CoT prompt with the format:

```
- Use the given sentences to create an example paragraph of an NLU task and its corresponding labels. The 5 sentences are: input.

- Make a plan then write and determine. Your output should be of the following format:

- Plan:

- Your plan here.

- Paragraph:

- Your paragraph here.

- Task:

- [Only the task name here, without additional information.]

- Labels:

- [Only the labels here, without additional information.]
```

The GPT will generate the corresponding answers in the defined format. To evaluate the generated task plans, we prompt the GPT again with the score prompt in the format:

```
- Analyze the following paragraph, then at the last line conclude "Thus the coherency score is s", where s is an integer from 1 to 10.
```

We filter out the generated texts with a score lower than 7. The additional data will be generated by the GPT with the selected proposals as prompts.

4.2 Pretraining

We then trained a RoBERTa model with the extended dataset using `RobertaForMaskedLM` provided by the `huggingface` library [4], which uses the default settings of `RobertaConfig` library and is also the same settings as the hyperparameter of the baseline provided by the organizers. In the training phase, we trained 5 epochs using the Trainer provided by the `huggingface`. We refer §C for detailed hyperparameters in Appendix.

4.3 Benchmarks and Evaluation

We evaluated the model using the evaluation pipeline tools[5] also provided by the organizer (Warstadt et al., 2023; Gao et al., 2021). This tool automatically performs experiments on 4 benchmarks:

1) Benchmark of Linguistic Minimal Pairs (BLiMP) (Warstadt et al., 2020a);

2) BLiMP Supplement[6], including Hypernym, QA Congruence Easy, QA Congruence Tricky, Subject Aux Inversion, and Turn Taking datasets;

3) General Language Understanding Evaluation (GLUE) (Wang et al., 2019), and

[3]The full datasets could be downloaded here: https://github.com/babylm/babylm.github.io/raw/main/babylm_data.zip

[4]https://huggingface.co/docs/transformers/model_doc/roberta

[5]https://github.com/babylm/evaluation-pipeline

[6]The relevant paper for this benchmark had not been published at the time of this project, and the relevant data can be found here https://github.com/babylm/evaluation-pipeline/blob/main/filter_data.zip

4) Mixed Signals Generalization Set (MSGS) (Warstadt et al., 2020b).

The detailed documentation of each benchmark can be found in §D. The organizer (Warstadt et al., 2023) also provided 3 models as baselines, including OPT-125M, RoBERTa-base, and T5-base, trained on the `babylm_100M` data.

5 Results

We compare the performance of our BabyLM (trained in the RoBERTa way) to the original RoBERTa-base (baseline). Table 1 shows our selected experimental results with: i) performance improvement by at least 3 points (+3), and ii) performance reduction over 3 points (-3). We report the performance with absolute performance difference of our BabyLM over baseline on the selected tasks, as well as the overall performance of the whole tasks. The full results are available in §D.

Tasks	Models		Diff.
	Ours	Baseline	
BLiMP			
Filler Gap	**78.52**	68	**10.52**
Sub.-Verb Agr.	**85.17**	76.2	**8.97**
Arg. Structure	**78.06**	71.3	**6.76**
Det.-Noun Agr.	**97.75**	93.1	**4.65**
Anaphor Agr.	**93.61**	89.5	**4.11**
Ellipsis	77.02	83.8	-6.78
Island Effects	45.85	54.5	-8.65
BLiMP Supplement			
Sub. Aux Inversion	**77.73**	45.6	**32.13**
QA Cong. Easy	**62.5**	34.4	**28.1**
Turn Taking	**62.5**	46.8	**15.7**
GLUE			
BoolQ	**65.84**	59.9	**5.94**
MNLI	**73.73**	68.7	**5.03**
MNLI-mm	74.76	78	-3.24
QNLI	76.86	82.3	-5.44
RTE	45.45	51.5	-6.05
AVG. (overall)	**73.95**	71.75	**2.2**

Table 1: Selected results of our BabyLM and the RoBERTa (baseline), where the performance of BabyLM improved by at least 3 points (in **bold**), or reduced (-) over 3. The metric in this table is all accuracy score.

We noticed that on the BLiMP benchmark, 5 indicators increased by more than 3 points, namely Filler Gap (+10.52), Subject-Verb Agreement (+8.97), Argument Structure (+6.76), Determiner Noun Agreement (+4.65) and Anaphor Agreement (+4.11), while two tasks dropped by more than 3 points, namely Ellipsis (-6.78) and Island Effects (-8.65). The average performance on this benchmark has also increased by 2.24.

On the GLUE benchmark, BoolQ and MNLI showed an improvement of 5.94 and 5.03, respectively, while MNLI-mm, QNLI, and RTE showed a decrease of -3.24, -5.44, and -6.05 respectively.

On top of that, we surprisingly find significant improvements in 3 tasks of the BLiMP Supplement benchmark: Subject Aux Inversion (+32.13), QA Congruence Easy (+28.10), and Turn Taking (+15.70). The average performance on this benchmark improved by 14.85 points.

The overall average performance is increased by 2.2, which shows that our model, pretrained with our reinterpreted small data, already demonstrates a great improvement.

6 Discussion

6.1 Augmented Dataset via CoT Prompting

We generated our data via the above-mentioned CoT prompting and the GPT-3.5-turbo for nearly 700, 000 lines, we show a case study of a part of the generated data here.

```
- Paragraph:
    - We have a few topics to cover in this paragraph. Firstly, a possible I.D. has been found in one of Gina's snapshots. Secondly, there is a new technology in development called autostereoscopic 3D that will allow people to watch 3D movies without glasses. This is great news for those who find wearing 3D glasses uncomfortable and causes eye strain. Unfortunately, the narrator regrets not asking Jean for the details about something. Lastly, the police are seen moving down the main street of Atenco, and we are tracking their movements.
- Plan:
-   1. Introduce the topic of the paragraph
    2. Mention the possible I.D. from Gina's snapshots
    3. Talk about the new technology called autostereoscopic 3D
    4. Mention the difficulty of wearing 3D glasses
    5. Mention the regret of not asking Jean for details
    6. Talk about the police and their movement down the main street of Atenco
```

```
- Task:
    - Text Classification
- Labels:
-   1. I.D. Mentioned
    2. Technology Mentioned
    3. Regret Expressed
    4. Police Mentioned
```

As we can see from the script, the paragraph is an extension of the input sentences sampled from the original dataset, while the plan and labels generated by the language model are the outlines, where the scenes also are the critical information from the generated paragraph. It means that our approach augmented the original data with interpretation, emphasis, and simplification, with which the model is possible to learn about a story with different versions and sizes and finally get a clearer understanding.

6.2 Performance in QA Congruence Easy

We analyzed the most noticeable improvement of the QA Congruence Easy dataset from the BLiMP Supplement benchmark, and dived deep into each case. This dataset consists of 64 single-choice questions with 20 *what*-questions, 25 *who*-questions, and 19 *where*-questions. Each question contains a question mark, and each answer ends with a period. Each question corresponds to 2 candidate answers, and the boundary of the candidate answers is clear, i.e., for the *what*- and *who*-questions, the answers contain an inanimate or an animate, and for the *where*-questions the answer is a location or a noun phrase. Obviously, the answer to the what-questions should be inanimate, like *a car*, the answer to the who-question should be animate, like *a doctor* or person's name *Sarah*, and the answer to the where-question should be location, like *at home*. The model is expected to select the answer that matches the question. For example, a question is "*Who did you see?*" and the candidate answers are 1. "*A doctor*", 2. "*A car*", and it is clear that the answer should be "*A doctor*". The final metric for the evaluation is accuracy.

6.2.1 Influence of the 3 Types of Questions

In these three kinds of questions, our model is better at answering the *what*-questions, where the accuracy is 75. Besides, it obtains an accuracy of 64 for the *who*-questions, and 47 for the *where*-questions.

6.2.2 Influence of the 2 Types of Answers

We also note that there are two forms of the answers:

1) *sentence*, where the answer is a complete sentence that includes at least the verb, e.g. "*I sent the package to europe*";

2) *fragment*, where the answer is a single word or a simple phrase, and does not include the verb, e.g. "*a car*".

The form of the two candidates' answers to each question is consistent, i.e., both candidates' answers are either sentences or fragments. The dataset contains 27 question-answer pairs in the form of sentences (42%) and 37 cases in fragments (57%). We also counted the accuracy on the above two forms, where the accuracy is 77.78 for sentences and 51.35 for fragments. Additionally, we also counted the accuracy with the different forms of the three questions i.e. *what*-, *who*-, and *where*-questions. The accuracy of the sentence labels on the *what*-questions is 80, while the fragment is 70. The accuracy on the who-question with sentence answers was 71 and 61 with fragment answers. On *where*-questions, the tasks with sentence answers obtained an accuracy of 80, however, it was only 11 with the fragment answers. Thus we can observe that our model is better at deciding with complete answers rather than fragments.

6.2.3 Influence of the 3 Types of Dialogues

Besides, we also notice that there are three types of dialogues for each question,

1) *direct* dialogues, where the question is started by a question word directly and the answer is direct with the answer, e.g., question: "*What did you get?*", candidate answers: "*I got a chair*", "*I got a doctor*";

2) *A-B* dialogues, where the letters *A* and *B* are used as names for both sides of the conversation before proposing the question and the candidate answers respectively, e.g. question "*A*: What did you sell?", candidate answers: "*B*: A chair.", "*B*: A doctor.";

3) *David-Sarah* dialogues, the person's name *David* is used as the questioner's name before the question, and *Sarah* is used as the answerer's name before the answer.

The dataset comprises 21 direct dialogues (32%), 22 *A-B* dialogues (34%), and 21 *David-Sarah* dialogues (32%), with the model's accuracy consistently ranging between 61-63% across these types.

We then explored the proportionality between these three forms of dialogue and the three kinds of questions. Of the 20 *what*-questions, 7 are written in *direct* dialogues, 6 are in *A-B* dialogues, and 7 are *David-Sarah* dialogues. we notice a difference in the accuracy, where the accuracy with *direct* dialogues is 100, the *A-B* dialogues have an accuracy of 83, and the David-Sarah dialogues reached only 45.

Of the 25 *who*-questions, 8 *direct* dialogues obtained an accuracy only of 25, while 7 *A-B* dialogues gained 85 accuracy and the accuracy of the 10 *David-Sarah* dialogues is 80. Out of the 19 *where*-questions, the accuracy of the 6 direct dialogues is 66%, 33% of *A-B* dialogues are correct, and the accuracy of the 4 *David-Sarah* dialogues is 50%.

From the above results, we can see that our model is good at selecting answers from *direct* and *A-B* dialogues on the *what*-questions. In contrast, for the *who*-questions, our model is good at selecting animates from the *David-Sarah* dialogues and the *A-B* dialogues, but not good at selecting the animate from the direct dialogues. It might be positively affected by the presence of the person's name. In the *where*-questions, the form of dialogues has a more limited effect on the performance.

6.3 Performance in QA Congruence Tricky

We compared the performance on the QA Congruence Tricky dataset, on which we have a very similar performance (35) to the baseline model. It contains 165 tricky questions including *who-*, *where-*, *when-*, *why-*, and *how many*-questions, where the proportions of the *who-* and the *where*-questions are 15% and 16% respectively. The accuracy of the *who-* and *where*-questions are only 37 and 30 respectively, differ from the accuracies in the QA Congruence Easy dataset.

We also notice that, in this dataset, our model is better at selecting fragment answers rather than answers in the form of sentences, where the accuracy with fragments is 62, while the accuracy of the sentences is only 10. On both *who-* and *where*-questions, our model is better at finding the answer in the *David-Sarah* dialogues (55 and 45 respectively in accuracy), and the accuracies of both questions in the other two dialogue forms are under 30. Similar to the fact shown in the easy dataset, the presence of people's names probably provides a sign to the animate and thus influences the performance, especially on the *who*-questions.

We analyzed the questions-candidate answers pairs from the tricky dataset, where both the questions and the candidate answers are generally shorter, e.g., the question is "*Who ate?*", and the candidate answers are "*A teacher ate.*", and "*Pasta ate.*", where the question only contains the *wh-* word, a verb, and a question mark, and the candidate answers contain only a subjective and a verb. The answers in the form of fragments are even shorter, e.g. to a question "*Who cooked?*", the candidate answers are "*Sarah*", and "*A sandwich*".

Besides the questions being more varied and complex, this dataset is more tricky, because the context is short. The candidate answers written in sentences are generally very similar to the fragments with only an additional verb, where the verb has been mentioned in the questions, which means the form of sentence possibly doesn't provide additional information, but may confuse the model to understand the answers.

7 Conclusion

In this work, we proposed the CoThought pipeline for training a BabyLM at a small scale, combining the LLMs' productivity with the concept of a child's cognitive learning ability. We let the raw training data for the BabyLM be reformulated by the LLM's CoT prompting (i.e. let the teacher think) and then train a BabyLM in a pretraining fashion based on the newly structured data (i.e. let the child co-think and learn). We compare the performance results of our BabyLM to another vanilla pretrained LM RoBERTa and demonstrate that our model achieves higher performance in many tasks including linguistic, question and answer, especially congruence tasks. This suggests that data processed by LLMs based on their contextual reasoning is more natural and efficient in the learning process, just as text revised by experienced teachers in the school is more suitable for students to learn and understand. And when we use data restructured by LLMs, even in the case of small data volume, the model is able to achieve the effect of a model trained from a large amount of data, or to be even better.

Limitations

One limitation of our work is the exclusive use of a specific LLM for data generation. It would be insightful to explore how performance varies when using different LLMs to generate the pre-training data. Different LLMs may introduce variability and diversity in the generated data, which could influence the effectiveness of the pre-training process. This aspect, while not explored in our current work, presents a promising avenue for future research to understand the impact of various LLMs on data generation and subsequent model performance.

Another limitation of our work is that our primary focus is on data generation, leaving potential improvements or optimizations in this domain unexplored.

Additionally, our model training exclusively utilized the RoBERTa architecture. Other architectures, including causal language models and various transformer variants, also showed potential research value. Therefore, exploring our approach across a broader range of architectures and identifying pretraining methods most compatible with our generated data remains an important area for future research.

By acknowledging these limitations, we hope to spur further research in this area, encouraging the exploration of data generation techniques, model architectures, and extended data methods in the context of small-scale language modeling.

Ethics Statement

This research was conducted in accordance with the ACM Code of Ethics. The datasets that we use are publicly available (Warstadt et al., 2023). We report only aggregated results in the main paper. We have not intended or do not intend to share any Personally Identifiable Data with this paper.

Acknowledgements

We thank the anonymous reviewers and the organizing committee for their efforts and helpful advice. E.N. was supported by MCML and CSC.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7:8.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *CoRR*, abs/2005.14165.

Joan Bybee. 2001. *Phonology and Language Use*. Cambridge Studies in Linguistics. Cambridge University Press.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Stephen L Chew and William J Cerbin. 2021. The cognitive challenges of effective teaching. *The Journal of Economic Education*, 52(1):17–40.

Noam Chomsky. 1957. *Syntactic Structures*. The Hague: Mouton and Co.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics.

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.

Emil Gegov, Fernand Gobet, Mark Atherton, Daniel Freudenthal, and Julian Pine. 2014. Modelling language acquisition in children using network theory. In *European Perspectives on Cognitive Science*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4849–4870, Toronto, Canada. Association for Computational Linguistics.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7, pages 785–794.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. 2020. A review of winograd schema challenge datasets and approaches. *arXiv preprint arXiv:2004.13831*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

HJ Levesque. 2011. The winograd schema challenge. aaai spring symposium: Logical formalizations of commonsense reasoning. *Palo Alto CA*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mitja Nikolaus and Abdellah Fourtassi. 2021. Modeling the interaction between perception-based and production-based learning in children's early acquisition of semantic knowledge. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 391–407, Online. Association for Computational Linguistics.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models.

Geoffrey K Pullum and Barbara C Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):9–50.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. *OpenAI*.

Matthew Saxton. 2010. *Child Language: Acquisition and Development*. Sage Publications.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

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

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 217–235. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference for Learning Representation (ICLR)*.

A Code and Model

The code for data processing and model training is available at: `https://github.com/oooranz/Baby-CoThought`.

Our BabyLM is available at: `https://huggingface.co/yaanhaan/Baby-CoThought`.

B Pretraining Data Statistics

The generated dataset for LM pretraining is available at: `https://huggingface.co/datasets/yaanhaan/Baby-CoThought-Data`.

We present a statistical analysis of the generated dataset. Given that our task revolves around creative NLU example generation, the dataset inherently encompasses a wide variety of tasks. This diversity is reflective of the creative nature of the task, allowing for a richer and more comprehensive pretraining process. Each example in the dataset includes an NLU example and its corresponding reason.

We plot the task distribution of the pretraining dataset in Figure 2. Tasks that appeared only once in the dataset are categorized as others.

The average number of words in the paragraphs across all examples in the dataset is approximately 115.25 words.

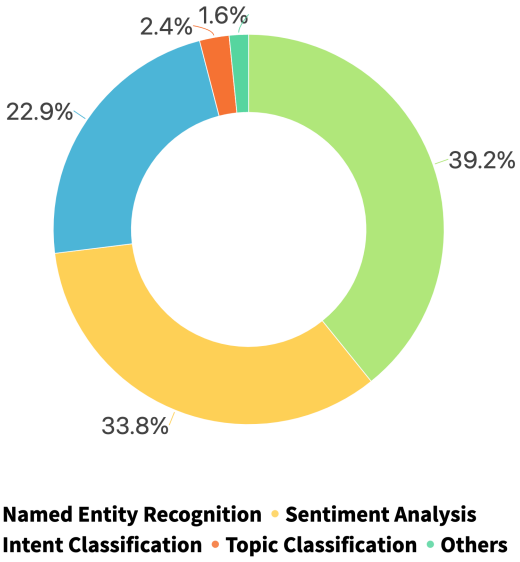

Figure 2: The distribution of the different NLU task examples in the pretraining dataset.

C Hyperparameter

We followed the instruction[7] and trained the tokenizers separately for the original dataset and our enhanced dataset via the `ByteLevelBPETokenizer` library with the hyperparameters shown in Table 2. Other hyperparameters were set to default and can be found in the document [8].

Hyperparameter	Value
vocab_size	52000
min_frequency	2
special_tokens	<s>, <pad>, </s>, <unk>,<mask>

Table 2: Hyperparameters used for tokenizers

Besides, we report our hyperparameters during the pretraining of our RoBERTa models in Table 3. We used the default settings from the `RobertaConfig` library. More default values and technical details can be found in the documents 3111[9].

Additionally, the evaluation process was done automatically via the evaluation tool provided by the organizer, without changing the hyperparameters, which can be found on the webpage [10].

[7] https://huggingface.co/blog/how-to-train

[8] https://github.com/huggingface/tokenizers/blob/main/bindings/python/py_src/tokenizers/implementations/byte_level_bpe.py

[9] https://huggingface.co/docs/transformers/model_doc/roberta#transformers.RobertaConfig

[10] https://github.com/babylm/

Hyperparameter	Value
attention_probs_dropout_prob	0.1
bos_token_id	0
classifier_dropout	null
eos_token_id	2
hidden_act	gelu
hidden_dropout_prob	0.1
hidden_size	768
initializer_range	0.02
intermediate_size	3072
layer_norm_eps	1.00E-12
max_position_embeddings	512
model_type	roberta
num_attention_heads	12
num_hidden_layers	12
pad_token_id	1
position_embedding_type	absolute
torch_dtype	float32
transformers_version	4.17.0
type_vocab_size	1
use_cache	TRUE
vocab_size	52000

Table 3: Hyperparameters used for pretraining

D Full Results

We used 4 benchmarks:

1) Benchmark of Linguistic Minimal Pairs (BLiMP) (Warstadt et al., 2020a), including Anaphor Agreement, Argument Structure, Binding, Control Raising, Determiner Noun Agreement, Ellipsis, Filler Gap, Irregular Forms, Island Effects, NPI Licensing, Quantifiers, and Subject Verb Agreement;

2) BLiMP Supplement[11], including Hypernym, QA Congruence Easy, QA Congruence Tricky, Subject Aux Inversion, and Turn Taking;

3) General Language Understanding Evaluation (GLUE) (Wang et al., 2019), including CoLA (Warstadt et al., 2018), SST-2 (Socher et al., 2013), MRPC (F1) (Dolan and Brockett, 2005), QQP[12] (F1), MNLI (Williams et al., 2018), MNLI-mm, QNLI (Levesque, 2011), RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), BoolQ (Clark et al., 2019), MultiRC (Khashabi et al., 2018) and WSC (Kocijan et al., 2020);

4) Mixed Signals Generalization Set (MSGS) (Warstadt et al., 2020b), including Control Raising Control (CR Control), Lexical Content The Control (LC Control), Main Verb Control (MV Control), Relative Position Control (RP Control), Syntactic Category Control (SC Control), Control Raising Lexical Content The (CR LC), Control Raising Relative Token Position (CR RTP), Main Verb Lexical Content The (MV LC), Main Verb Relative Token Position (MV RTP), Syntactic Category Lexical Content The (SC LC), Syntactic Category Relative Position (SC RP).

to process our evaluation.

The organizer provided three baseline models, including OPT-125M[13] , RoBERTa-base[14] , and T5-base[15]. We show our full results in Table 4.

evaluation-pipeline#hyperparameters

[11] https://github.com/babylm/evaluation-pipeline/blob/main/filter_data.zip

[12] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[13] https://huggingface.co/facebook/opt-125m

[14] https://huggingface.co/roberta-base

[15] https://huggingface.co/t5-base

Tasks	Models				Difference	
	Ours	OPT-125m	RoBERTa-base	T5-base	in abs	in rel.
BLiMP						
Anaphor Agreement	93.61	**94.90**	89.50	66.70	4.11	4.59%
Argument Structure	**78.06**	73.80	71.30	61.20	6.76	9.48%
Binding	72.84	**73.80**	71.00	59.40	1.84	2.59%
Control Raising	69.55	**72.20**	67.10	59.80	2.45	3.65%
Determiner Noun Agreement	**97.75**	93.10	93.10	53.80	4.65	4.99%
Ellipsis	77.02	80.50	**83.80**	49.10	-6.78	-8.09%
Filler Gap	**78.52**	73.60	68.00	70.00	10.52	15.47%
Irregular Forms	**91.25**	80.80	89.60	75.50	1.65	1.84%
Island Effects	45.85	**57.80**	54.50	43.60	-8.65	-15.87%
NPI Licensing	**67.35**	51.60	66.30	45.60	1.05	1.58%
Quantifiers	70.58	**74.50**	70.30	34.20	0.28	0.40%
Subject Verb Agreement	**85.17**	77.30	76.20	53.20	8.97	11.77%
BLiMP Supplement						
Hypernym	49.07	46.30	50.80	**51.10**	-1.73	-3.41%
QA Congruence Easy	62.50	**76.50**	34.40	45.30	28.10	81.69%
QA Congruence Tricky	34.55	**47.90**	34.50	25.50	0.05	0.14%
Subject Aux Inversion	**77.73**	85.30	45.60	69.20	32.13	70.46%
Turn Taking	62.50	**82.90**	46.80	48.90	15.70	33.55%
GLUE						
CoLA	74.09	73.70	75.90	**76.30**	-1.81	-2.38%
SST-2	**88.78**	86.60	88.60	88.00	0.18	0.20%
MRPC (F1)	80.45	82.10	80.50	**85.90**	-0.05	-0.06%
QQP (F1)	**81.20**	77.80	78.50	79.70	2.70	3.44%
MNLI	**73.73**	70.10	68.70	71.50	5.03	7.32%
MNLI-mm	74.76	71.90	**78.00**	74.00	-3.24	-4.15%
QNLI	76.86	80.10	82.30	**83.10**	-5.44	-6.61%
RTE	45.45	**67.70**	51.50	60.60	-6.05	-11.74%
BoolQ	65.84	66.00	59.90	**69.00**	5.94	9.91%
MultiRC	62.21	61.10	61.30	**62.40**	0.91	1.49%
WSC	**61.45**	59.00	61.40	60.20	0.05	0.07%
MSGS						
CR (Control)	83.96	**97.20**	93.00	95.10	-9.04	-9.72%
LC (Control)	94.49	82.60	**100.00**	**100.00**	-5.51	-5.51%
MV (Control)	99.98	**100.00**	**100.00**	**100.00**	-0.02	-0.02%
RP (Control)	**100.00**	99.80	**100.00**	99.80	0.00	0.00%
SC (Control)	88.44	88.10	**89.00**	88.70	-0.56	-0.62%
CR LC	67.07	75.30	68.30	**76.70**	-1.23	-1.80%
CR RTP	**70.71**	67.10	66.80	69.40	3.91	5.86%
MV LC	66.61	66.30	66.60	**67.00**	0.01	0.01%
MV RTP	67.59	66.80	**80.20**	67.70	-12.61	-15.72%
SC LC	75.47	**84.80**	67.40	72.70	8.07	11.98%
SC RP	**70.90**	62.00	67.40	68.00	3.50	5.19%

Table 4: Full results, with difference of our BabyLM over RoBERTa-base (baseline). Metric of MRPC and QQP from GLUE is F_1, in other tasks the metric is accuracy. The best results of the four models are marked in **bold**.