COMP 551 Mini-Project 4
Reproducibility in Machine Learning

Yang Kai Yam, Xu Michael, and Tian Yu Hao

School of Computer Science, McGill University

Abstract

Our project revisits a 2023 study on training smaller language models (SLMs) for proficient language generation. The original work first presented the challenge of achieving fluency and coherence in text generation, which are still characteristics mainly attributed to large-scale models. Then it posits that by training the models on specialized text data stripped to its bare essentials, we can have much smaller models (ones with only a few million parameters) to be as good or even better than "state-of-the-art models" for text generation that is grammatically correct, diverse, and demonstrate decent reasoning capabilities [1]. Our replication efforts indicated that while models trained on TinyStories showed commendable performance for their size, they still lacked the precision and quality demonstrated in the original paper. We also trained one SLM on 2.2 million stories in 24 hours using a single V100 GPU, which is good enough to generate desirable completions.

Introduction

Language as we know it is a multi-faceted concept that one would need to know about more than just the words themselves to be able to use in an everyday setting. Grammar, syntax, lexicography, logic, context-tracking and so much more is required to truly grasp the inner workings of text. Recent developments in large language models (LLM), capable of generating coherent English text, hint at these models' deeper linguistic understanding.

However, the scale at which these abilities emerge in language models remains uncertain. It is observed that smaller language models like GPT-Neo and GPT-2, despite their 100 million parameter size, often fail in coherent sentence formation, irrespective of their training. This problem could stem from the "intrinsic complexity of natural language" [1] or even the expansive diversity of training corpora. A hypothesis is that smaller models, when trained on extensive datasets like Wikipedia, might get overwhelmed and take a hit to their learning efficiency.

Our study, as did the original paper, seeks to understand whether training smaller models on a more refined dataset can enhance their performance to be on par with LLMs. We followed the same approach as outlined by Eldan and Li, employing models with various sizes, ranging from 1M to 33M parameters. These SLMs were tasked with completing stories, using prompts derived from the same TinyStories dataset to ensure maximum reproducibility. And for the sake of consistency, our baseline model was GPT-2XL (1.5B parameters) as well.

Initial runs on the example storytelling prompts given in the paper (Section D of Results) indicate decent performance from the scores attributed by GPT 3.5 to grammar, creativity, and consistency. And from looking at the performance of various models on contextual prompts we see that TS trained models remain pretty consistent throughout. All our results seem to be in line with the original paper, although not as high-quality as the ones outlined there.

All in all, the experiments are decently reproducible and verifiable, given that, the original conclusion holds about light SLMs.

Dataset

The general approach to training examples found in the TinyStories dataset was explained in the Introduction. More formally though, it is a corpus that has the basic components of grammar, vocabulary, facts, and reasoning, but is more restricted in terms of content compared to the general LLM training corpus. One important thing to note is that in the original paper, the intuition behind the formation of such a dataset was based on the vocabulary that a normal 3 year old child would understand. As for the text itself, it was generated by GPT-3.5 and GPT-4. The number one thing one has to circumvent in using such a text generation method is the lack of diversity problem. Elden and Li managed to avoid it by choosing specific vocabulary in a list of about 1500 words, and for each generation to use at least 3 different words split into noun-verb-adjective while incorporating a certain story feature into the generating prompt (such as a dialogue, plot twist, good or bad ending).

Results

A. Test dataset and Evaluation Method (GPT-Eval)

Follow the same method from original paper. Using 50 stories from validation dataset and keep the first half as input prompts. We choose the 2nd stories until the 51th one to reach a better generalization because they share the same beginning "Once upon a time...".

During the evaluation, we had the model complete a story for each prompt and used GPT-3.5 to obtain average scores for 50 prompts. We observed identical completions from the Tinystories models using the same prompts, so the additional 9 completions mentioned in the original paper were no longer necessary. Due to a limited budget for evaluation with commercial GPT, we made several modifications from the original paper. We used GPT-3.5 for evaluation instead of GPT-4. We also compared the Tinystories models with GPT-3.5 instead of GPT-4. And we only completed each prompt once, as opposed to generating 10 completions as in the original paper.Similar to the original paper, we used a temperature one across all models in the performance comparison. .

B. Performance Comparison Among Models

Generally, larger models perform better across the three scores, showed in figure 1 and figure 2. Compared to GPT-3.5 with 175 billion parameters, much smaller models achieve comparable performance in terms of generating hundreds of words in stories. From the figure 2, we can see that all four models with more than 20 million parameters are very close to GPT-3.5 in creativity and consistency scores. However, GPT-3.5 slightly outperforms them in grammar scores, which may indicate that these models need a more complex structure to correct small grammatical errors.

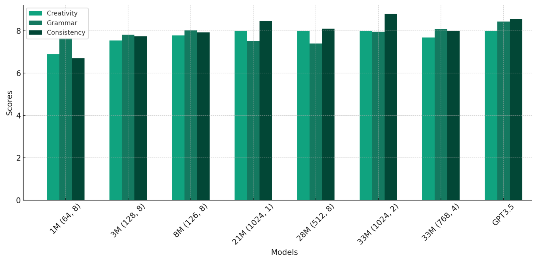


Figure 1: *Scores (format: model name, hidden size, layer)*

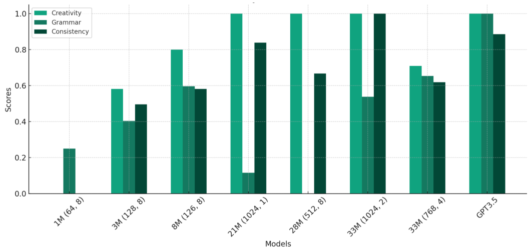


Figure 2: *Normalized Scores according to (score - $score_{min})/(score_{max}) - (score_{min})$*

C. Analyze with Storytelling Prompts

To gain a deeper understanding of the capabilities of various models, we analyzed their performance in story-telling scenarios using specific example prompts (Appendix 1). We assessed the responses from the baseline model alongside various configurations of the TinyStories SLM, created by GPT-Eval. These were judged based on grammar, creativity, and their ability to maintain continuity with the story's initial direction, with the specific prompts used for this evaluation also located in Appendix 2.

Figure 3 illustrates the comparative performance of these models. In the chart, "TS" represents the SLM trained using TinyStories, varying in parameter count. Models TS-33M and TS-28M, despite having substantially fewer parameters

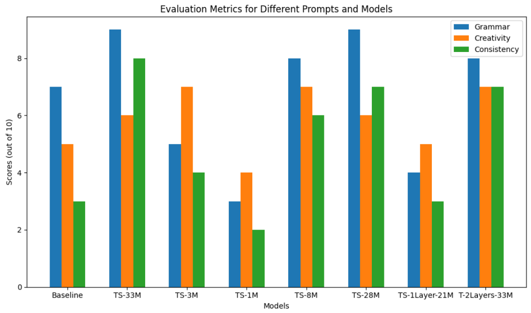


Figure 3: *The Performance of different LM testing with the Prompt 1 shown in Appendix 2*

than GPT-2's 1.5B, show remarkable efficiency. Their performance often matches or exceeds that of GPT-2 in producing coherent narratives. Both TS-33M and TS-28M display consistent output across various prompts, showcasing their stability and dependability in text generation. Notably, TS-33M, with its four transformer layers, outshines TS-2Layers-33M, which has just two layers, underscoring the benefit of additional layers for enhancing sentence coherence.

On the other hand, smaller models like TS-1M and TS-3M, with their limited parameters, generally fall short in text generation, highlighting the significance of model size in such tasks. Interestingly, TS-8M, with a modest parameter count of 8M, still manages to produce reasonably coherent sentences, suggesting that smaller models can still be effective in text generation to some extent. In contrast, GPT-2 demands considerably more time for operation compared to these smaller models, a factor that could be vital in scenarios requiring real-time application.

D. Analyze with Themed Prompts

Experimenting with the TinyStories dataset-trained models on themed prompts was truly where the most interesting results could be found. By "themed-prompts", we mean prompts that are specially engineered so that they test a model's certain capability be it context-tracking, reasoning, or knowledge of fact. Presented in Table 1 is the colour-coded table for the generations of the various models on six different context-tracking prompts, each with their built-in logic to keep in mind.

In general, we see that the TS-1M, 3M, and 8M models performed the worst out of the bunch, hinting at the importance of emdedding dimensions. That is to say, the more parameters a model has, the more accurately it can track context. However, this is already a pretty-well known notion. What is interesting here though, looking at the context table, it seems that the number of attention layers is crucial in maintaining consistency based on the context. For example, we see that TS-1L-21M performs much worse than we expect it to based on the number of parameters it possesses. If anything, it performs about as good as TS-1M. And theoretically, that tracks: the more layers of global attention a model has, the better it can keep track of context throughout.

Given that the tables for reasoning and factual knowledge prompts are slightly too big to include realistically in either our report or the Appendix, they will be included as standalone CSV files in our final submission. From there, we can see that indeed, as the original paper outlines: "the embedding dimension is more crucial for the embedding dimension is more crucial for capturing the meaning and the relations of words, the number of layers is more crucial for capturing long-range dependencies in the generation" [1].

One last thing to remark from our work in this section is the fact that SMLs can be highly unstable depending on the max length of generation and parameters controlling the diversity/creativity of the model. One thing that happened very often during evaluation is that of TS models on the themed prompts were blank or completely non-sensical generations. In a further study, it would be interesting to find out whether it's possible to create a setup for stable generations.

Prompt	TinyStories-33M	TinyStories-3M	TinyStories-1M	TinyStories-8M	TinyStories-28M	TinyStories-1Layer-21M	TinyStories-2Layers-33M	GPT2XL
’Hi Jane, have you seen Alice? I can’t find her anywhere,’ said Jack.	🟩 Jane smiled and said, "I know where you can find her. I’ll help you look." [...] they couldn’t find Alice.	🟨 Jane was very sad. She wanted to help Jack, but she was too scared to ask. [...]	🟥 Jane was so excited. She opened her eyes and saw a big, shiny object.	🟥 [...] Jack said, "I’m sorry, Jane. I can’t find you anywhere. I’m too busy looking for you."	🟩 Jane smiled and said, "I’m sure she’s here somewhere. Let’s go look together." [...]	🟥 Alice was so excited and said, ’Let’s go to the park!’ Alice and Jack went to the park. [...]	🟩 Jane was so excited to find her friend. She ran to the park and started to look for Alice. [...] But she couldn’t find her friend.	🟩 ’I’m sure she’s in the garden,’ said Alice. ’I’ve been looking all round the garden all day, and I can’t see her anywhere.’ ’I wish I could see her,’ said Jack.
Max had two dogs. [...] He told the kid, ’I see you have a Brown dog. I also have two dogs.’	🟨 The kid said ’Yes, I see you. That’s a nice dog.	🟨 The kid said, "That’s a nice dog.	🟥 Max was very happy. He said, "Hi, Mr. Lee. I’m a dog.	🟥 The kid smiled and said, ’Yes, they are Brown. They are my favorite.	🟩 The kid smiled and said, ’That’s great! I’m so happy to meet you.	🟥 The kid was very happy to meet Brownie. Max and the kid walked together to the park.	🟨 The kid said, ’That’s a nice dog.	🟥 The kid said, ’I have a white dog.’ Max said, ’I have two dogs.
Anne had a piece of candy in her left pocket[...] ’Anne, what is that you have in your left pocket?’	🟩 Anne replied, ’It’s a candy, Mommy.	🟨 Anne said, ’I’m sorry, Mom. I didn’t know it was yours.	🟩 Anne had a piece of candy in her left pocket and a piece of chocolate in her right pocket.	🟩 Anne said, ’It’s a candy, Mommy!’ Anne’s mom smiled and said, ’That’s a very special candy, Anne.	🟩 Anne had a piece of candy in her left pocket and a piece of chocolate in her right pocket.	🟥 Anne was so excited and she quickly put the piece of chocolate in her pocket. Anne was so happy that she had found the chocolate.	🟥 Anne replied, ’It’s a piece of chocolate, Mommy. I’m going to keep it safe in my pocket.	🟩 Anne said, ’I have a piece of candy in my left pocket and a piece of chocolate in my right pocket.
Alice had both an apple and a carrot in her bag. [...] She reached into the bag again and took out the carrot.	🟩 She gave it to Jack and said, "Here you go, Jack. This is for you." Jack took the carrot and bit into it.	🟥 She was so happy and thanked Jack for his help. Jack smiled and said, "You’re welcome, my sweet girl.	🟥 She was so happy that she had found the perfect apple.	🟩 She gave it to Jack and he smiled. Alice said, "Let’s share this apple and carrot.	🟩 She gave it to Jack and said, "Here, you can have this carrot. It’s yummy!" Jack was so happy.	🟥 Jack was so happy that he hugged Alice and said, "Thank you for the apple!" Alice smiled and said, "You’re welcome.	🟩 Jack was so happy to have his apple. He thanked Alice and they both enjoyed their snack. The end.	🟥 She gave it to Jack. She reached into the bag again and took out the apple. She gave it to Jack.
Alice and Jack walked up the street and met a girl [...] ’Hi, I’m Jane. What are your names?’	🟩 Alice and Jack introduced themselves and said, ’We are Alice and Jack. We are friends.	🟥 Alice said, ’I’m Jane. I’m a girl. I’m a girl. I’m a girl. I’m so happy to meet you.	🟥 The girl said, "I’m Ben. I’m just playing with you. I’m going to play with you.	🟩 Alice and Jack said their names and they started to talk. They talked about the things they saw and the things they saw.	🟩 Alice and Jack said We’re Alice and Jack.	🟥 Alice said, ’I’m Lucy. I like to play dress up.	🟩 Alice smiled and said, ’My name is Alice. What’s your name?’ Jack said, ’My name is Jack.	🟨 ’Jack and Alice,’ said Jack. ’I’m Jane’s sister, Jane,’ said Alice.
Diva was hungry and wanted to bake a cake [...] She asked the squirrel, ’Would you happen to have some sugar?’	🟩 The squirrel said, ’Yes, I will give you some sugar if you help me bake a cake.	🟩 The squirrel said, ’Yes, I would love to help you.	🟥 The squirrel said, "No, I want to help you.	🟥 The squirrel said, ’Yes, I would love to have some.	🟩 The squirrel said, ’Yes, I do.	🟥 N/A	🟩 The squirrel said, ’Yes, I do. I have some sugar in my house.	🟩 The squirrel said, ’Yes, I have some sugar.

Table 1: *Context prompts annotated by color with respect to context-keeping capability. Note: some prompts are cut a bit short for the sake of space. Red = Nonsensical or not correct, Yellow = Has flaws but is on the right track, Green = Follows the prompt.*

E. Interpretability Analysis

The attention matrix shown in Figure 4 demonstrates that within the transformer layer, tokens primarily attend to themselves and their immediate neighbors, as evidenced by strong diagonal patterns. This self-attention mechanism is typical in language models, reflecting their focus on local context to predict the next word in a sequence. This local focus is crucial for the model to generate text that is coherent and immediately contextually relevant.

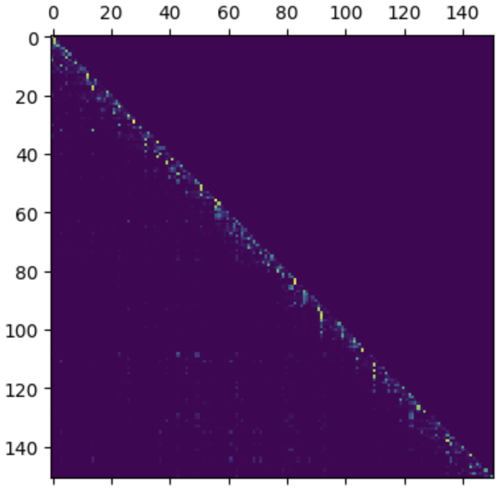


Figure 4: *The attention matrix for one head in one layer*

In contrast, the distinct patterns across different attention heads suggest specialized roles, with some focusing on nearby syntactic relationships and others on distant semantic connections. Off-diagonal bright spots in certain heads imply the model's ability to capture long-distance dependencies, crucial for maintaining narrative flow and understanding complex linguistic structures like the interplay between subjects and objects or verbs and their arguments.

F. Hyperparameter Tunning

A higher dropout rate negatively impacts performance, showed in figure 5. One possible explanation is that the dropout strategy combats overfitting, yet the model has not yet overfitted. According to the original paper, the evaluation loss consistently decreases after thousands of training steps, suggesting that two million training data points are insufficient to cause overfitting in the model. Furthermore, after altering the dropout rates, we trained the model on a manually constructed validation dataset, which does not overlap with the stories from the training set. Assuming the model is trained for one epoch or less, a common practice for large language models, overfitting is unlikely given the diversity of the synthetic story dataset. Therefore, it is reasonable to observe a decrease in performance when applying the dropout strategy to a model that has not overfitted.

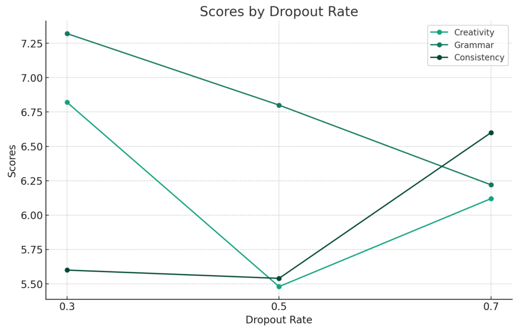


Figure 5: *Scores for different dropout rates*

H. Training GPT-Neo From Zero

Trained the model with one million parameters (removed pretrained weights) on the original paper's training dataset (2,119,719 stories). Used eight data points per step and only trained for one epoch. The original paper is not well-written in the training section because it does not provide explanations about the label setting using the training dataset. Given the nature of the autoregression model, we assume this model based on gpt-neo architecture also predicts the next word based on the one before it. Thus, we set labels equal to inputs during the tokenization process, enabling the model to use the word following the current word as labels. Similar to the original paper, we observed that the training loss reduces slowly after thousands of training steps in figure 6. Although the orginal paper does not explain how many data points are in one step, the trend in our image is similiar to their.

Discussion and Conclusion

The SLM can generate coherent texts The efficacy of small language models (SLMs) in generating coherent English narratives is largely

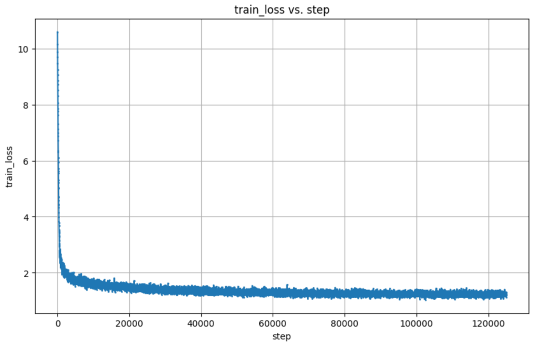


Figure 6: *Training loss for model with 1M parameters*

due to their focused training on the TinyStories dataset, which distills the language to its core components. This specialization enables the SLMs to master the essential structures of English with remarkable fluency and consistency, despite their limited parameter count. The simplicity of the dataset not only streamlines the model's learning process but also ensures that the generated content remains varied and contextually appropriate, showcasing the model's genuine understanding and learning of language patterns rather than mere memorization.

In addition, the architecture of these SLMs is optimized to capture linguistic essentials efficiently, suggesting that intelligent design can compensate for smaller sizes. These models excel at utilizing contextual cues, which allows for coherent text generation by focusing on the most relevant parts of a narrative. By effectively leveraging the immediate context and honing in on the fundamental skills necessary for text completion, SLMs demonstrate that a well-constructed model can achieve a high level of language proficiency, challenging the notion that larger models are always superior for complex language tasks.

Lower Computational Cost We found the response time from the large language model GPT2-XL is 10 times slower than the small language model trained by TinyStories. It highlights the lower computational cost of using SLMs, these models require significantly less computational resources, enabling faster and more efficient training. This efficiency contrasts with large language models that demand extensive computational power and time. The reduced cost of SLMs opens up opportunities for more widespread and diverse applications, especially in settings where computational resources are a constraint. The results are similar to our studied paper that an SLM can be trained in less than a day on a single GPU, making it accessible for environments with limited resources.

The Performance in Attention Matrix The findings from the experiment, such as diagonal patterns in attention heads, indicate a focus on immediate lexical context, typical in language processing. The variability among heads suggests functional specialization, where different heads are tuned to various linguistic aspects like syntax or semantics. Off-diagonal attention points to the model's ability to understand longer-range textual relationships, crucial for grasping complex sentence structures. These observations underscore the capability of small language models to efficiently capture essential linguistic features, despite their constrained size and computational simplicity.

Conclusion Our evaluation method deviated slightly from the original paper due to budget constraints. Instead of using GPT-4 for evaluation, we employed GPT-3.5. Interestingly, this did not significantly affect the outcome, as we observed similar story completions to those reported in the "TinyStories" study. This finding underscores the potential of SLMs in specialized applications, where the training dataset is tailored to include essential elements of natural language but within a more focused scope.

Statement of Contribution

Yang Kai Yam - Implemented the SLM model preparation and data preparation, util functions for experiments. Analyze the model performance with different prompts and interpretability analysis, and write their discussion.

Xu Michael - Implemented evaluation methods and performance comparison. Also trained a GPT-Neo model after removing the pre-trained weights.

Tian Yu Hao - Coded the section themed-prompts in the report. Helped the abstract, introduction, results, and conclusion sections of the report.

Reference

1. Eldan, R., & Li, Y. (2023). TinyStories: How Small Can Language Models Be and Still Speak Coherent English? arXiv:2305.07759. https://arxiv.org/abs/2305.07759

Appendix

1. The Prompt for evaluation (GPT-Eval)

In the following exercise, the student is given a beginning of a story. The student needs to complete it into a full story. The exercise tests the student's language abilities and creativity. The symbol *** marks the separator between the prescribed beginning and the student's completion:

[Original Sentence] *** [Extended Sentences]

Now, grade the student's completion in terms of grammar, creativity, and consistency with the story's beginning. Each metric is ranging from 0 to 10. Please provide the information as a number array only, without additional explanation or text.

(Appendix 1. The Evaluation Prompts for the results from LLM)

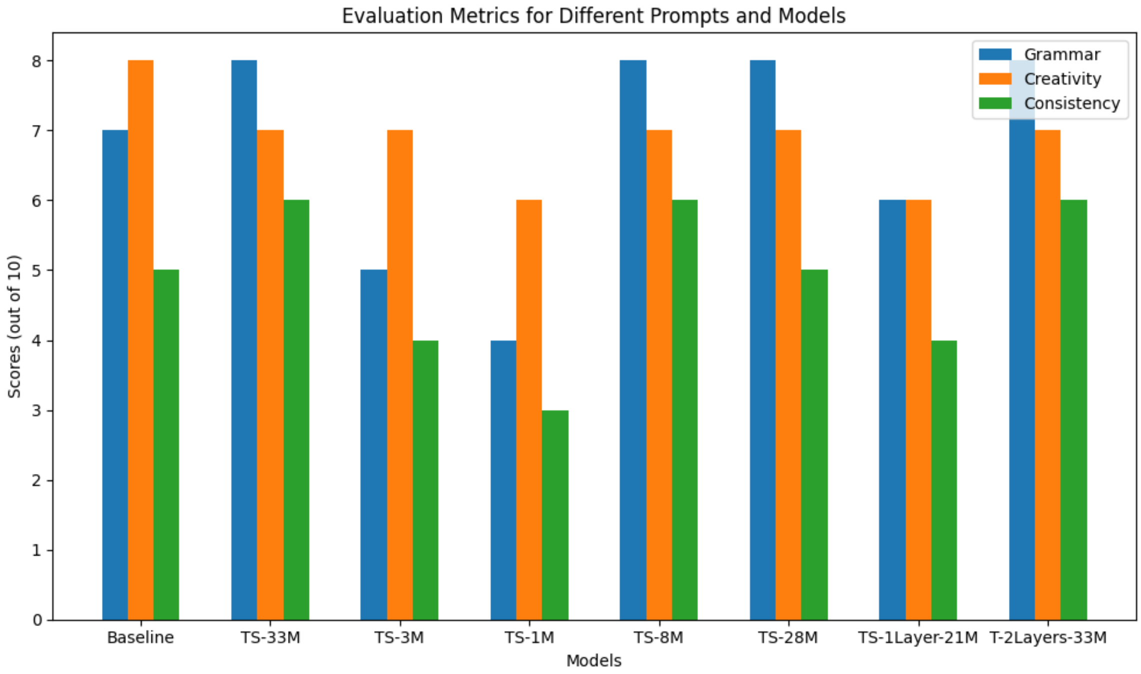
The prompt is used to evaluate the output performance from the baseline models and small language models. We are using the API for GPT3.5 and GPT4 to analysis the generated sentence in terms of creativity, grammar, and consistency with the story's beginning.

2. Testing With Story-telling Prompts

We have tested three different story-telling prompts to examine the ability of the model to complete the story, finally evaluating the generated context in terms of grammar, consistency, and creativity. Different structures of SLM model training with TinyStories are investigated. One of the prompt analyses is in the report Section D, and the other two analyses are shown here.

A. Example Prompt 1
Once upon a time there was a little girl named Lucy. She was very adventurous. She loved to explore the world around her, especially when it was bright and sunny outside. One day, while exploring the nearby park, Lucy came across a ladder leaning on a wall. She was curious to see what's on top, so she climbed the ladder, but when she reached the top, the ladder fell and she was stuck. A nearby park ranger noticed her and shouted out, "

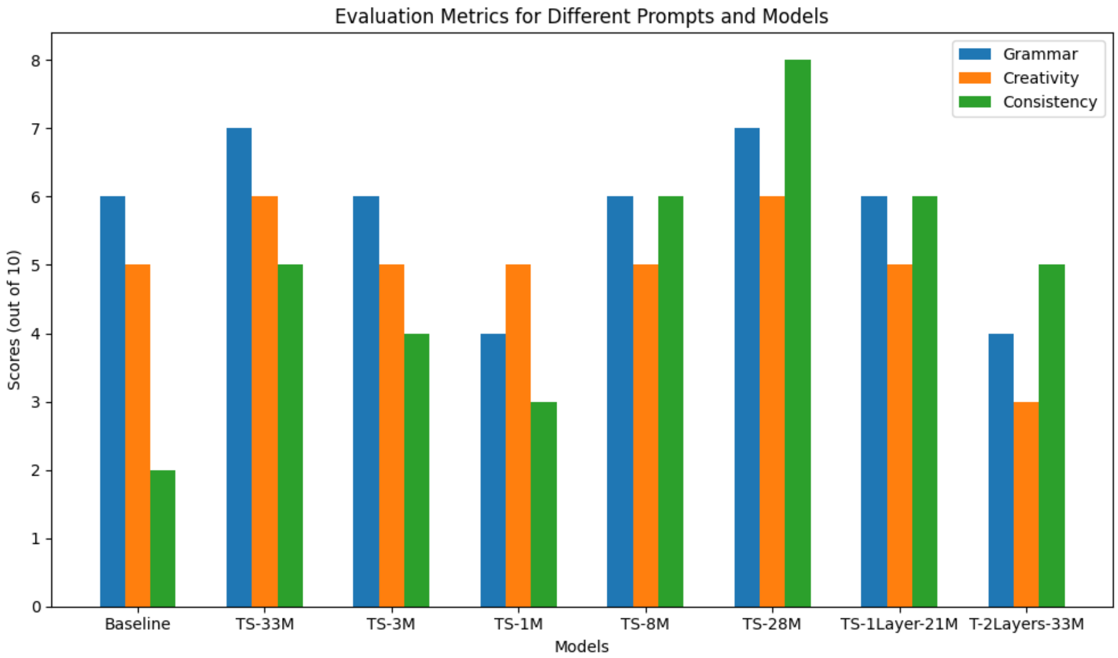
B. Example Prompt 2
Once upon a time there was a pumpkin. It was a very special pumpkin, it could speak. It was sad because it couldn't move. Every day, it would say



(Appendix 2A. The evaluation results for different LLM testing with Prompt 2)

C. Example Prompt 3

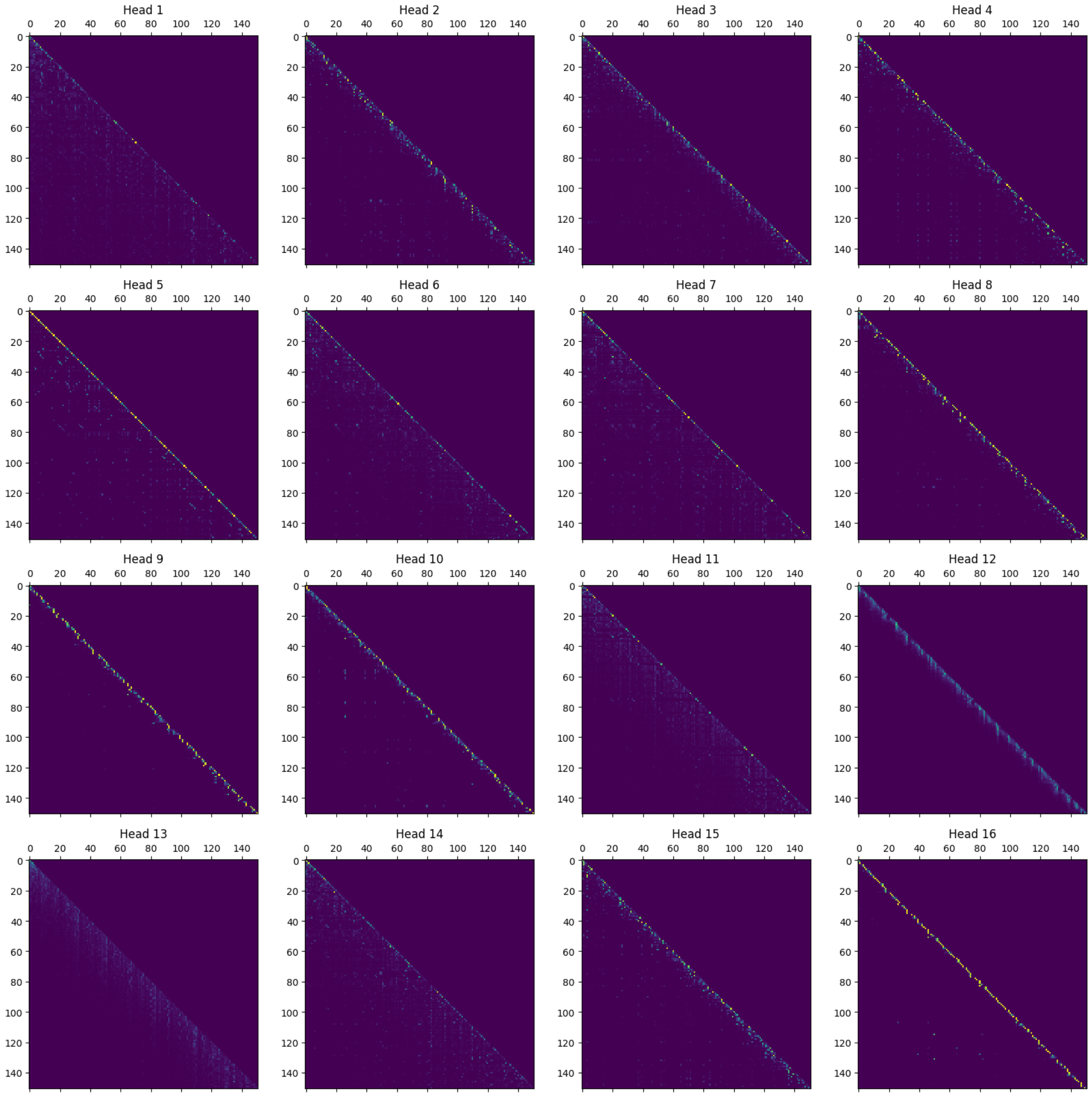
Once upon a time, there lived a black cat. The cat belonged to a little girl called Katie. Every day, Katie would take her cat for a walk in the park. One day, as Katie and her cat were walking around, they saw a mean looking man. He said he wanted to take the cat, to which she replied ”This cat belongs



(Appendix 2B. The evaluation results for different LLM testing with Prompt 3)

3. The Attention Matrix from one layer

Interpreting the role of different attention heads, there are 16 attention heads in one transformer layers



(Appendix 3. Multi-scale distance-based attention)