Pushing the Limits of what a Transformer can do - TinyStories

Team - NoLearningPossible
Ameya Rathod - 2022111021
Anish R Joishy - 2022111014
Aaditya Vardhan Narain - 2022111038

November 2024

1 Introduction of the Problem

Most state-of-the-art language models, such as GPT-3 and similar large language models (LLMs), are extremely large and computationally demanding. Their high resource requirements limit their accessibility and practical application in real-time or resource-constrained environments, such as embedded systems, mobile devices, or educational tools. These models' size and complexity also introduce substantial barriers to understanding and controlling language model behavior, especially in terms of narrative generation and other more specific tasks. Therefore, there is a growing need for language models that can perform robustly while remaining small and computationally efficient.

The TinyStories approach explores how small-scale language models, trained on carefully curated datasets of simple, short stories, can achieve remarkable fluency and coherence in language generation. The authors aim to show that it is possible to distill narrative competence into smaller models that still perform effectively in specific domains—like storytelling—despite lacking the broader capabilities of larger, general-purpose language models. The paper presents an innovative dataset of "TinyStories," which consists of short, child-friendly stories specifically designed for training and evaluating smaller language models. These stories are chosen for their simplicity, clear plot progression, and compact structure, making them an ideal foundation for evaluating the story generation capabilities of smaller models.

Through this targeted dataset, TinyStories demonstrates that with appropriate data and task-specific training objectives, smaller language models can perform surprisingly well in generating contextually appropriate, narrative-driven text. The paper's findings suggest that the inherent complexity required for certain NLP tasks, such as storytelling, may not be as high as previously assumed. Instead, smaller models can achieve impressive results with the right data and task-aligned training, effectively bridging the gap between model size and language generation quality. This research is significant as it opens new pathways for developing more efficient language models for applications where computational resources are limited, or where interpretability and control over the model's behavior are critical.

2 Selection of baselines

To evaluate the performance of the TinyStories models, the authors carefully selected a range of baselines that vary in size and architecture, comparing the storytelling quality and language fluency across models. This approach allowed them to rigorously assess how well TinyStories models perform relative to other language models, particularly focusing on coherence, grammatical correctness, and relevance to the input prompts.

The baseline models chosen in the paper span a range of model sizes, including both large-scale, state-of-the-art language models as well as smaller models commonly used in resource-constrained settings. Specifically, the authors compare the TinyStories models against GPT-2 and GPT-3 variants at different scales, recognizing that GPT-3 models set a high standard for language generation quality but come with immense computational and memory demands. By including these larger GPT models, the authors establish an upper bound for narrative quality, allowing them to benchmark TinyStories against some of the best available language generation systems.

For our project, we have considered GPT2-medium and GPT2-small as our baseline models. We did not use the higher models as they are not free to use. Also we are using Gemini-1.5-Flash model as the evaluator for both the baseline models and the small models that we have trained from scratch.

3 Dataset Characteristics

The **TinyStories** dataset is a collection of short, simple narratives designed to evaluate the storytelling abilities of Language Models (LMs). Each story is written in plain English, using simple vocabulary and clear grammar, making it in a structure of short kid's stories. The dataset features diverse themes, logical story structures with clear beginnings, middles, and ends, and covers a wide range of topics, from everyday life to imaginative adventures. Stories are concise, typically a few hundred tokens long, and suitable for benchmarking coherence, reasoning, and interpretability in AI-generated text. Its focus on simplicity and ethical content ensures it is free from offensive or harmful material, making it ideal for educational and creative AI applications. The dataset is synthetic and generated by GPT-4. Its train split has about 2.1M stories and validation split has about 22k stories.

The **TinyStoriesInstruct** dataset has the some words that must be contained in the stories, features of that story and a short summary for the story along with the story itself. This can be used to benchmark instruction following capabilities of LMs.

4 Evaluations

4.1 Model Evaluation Table

The table below summarizes the The performance of small models trained on TinyStories.

Hidden Size	Layers	Heads	Eval Loss	Creativity	Grammar	Consistency	Plot Sense
128	8	4	2.00	1.9	2.7	1.8	2.1
128	12	4	1.96	2.1	3.15	2.55	2.15
256	8	2	1.60	2.4	4.8	4.55	3.45
256	8	4	1.59	2.15	3.85	3.7	2.65
256	8	8	1.58	2.5	5.6	5.75	3.5
256	12	4	1.55	2.25	4.9	4.2	2.8
512	8	4	1.33	2.6	5.85	4.7	3.65
512	12	4	1.30	2.55	6.0	5.8	4.2

Table 1: Evaluation of Model Configurations on TinyStories Dataset

4.1.1 Variation across hidden size

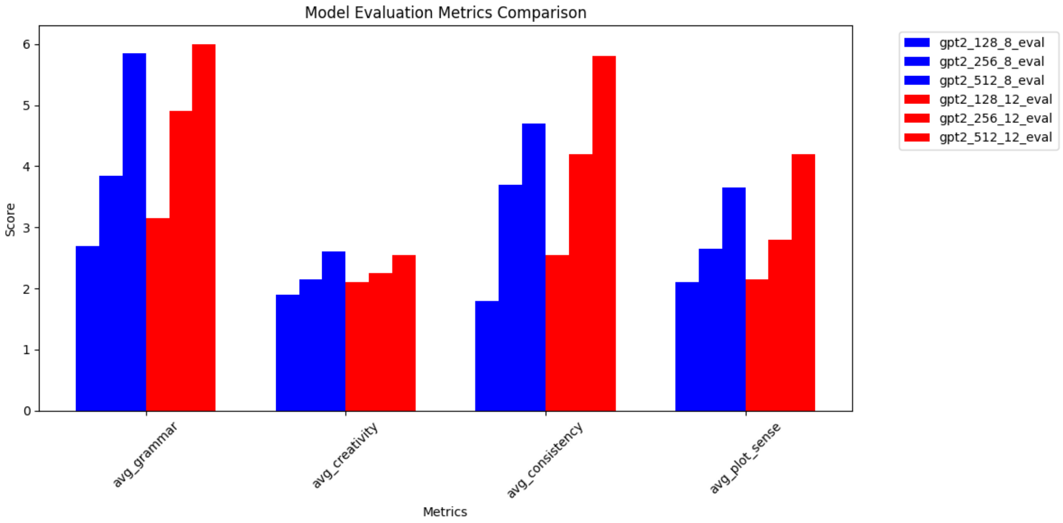


Figure 1: We can see that hidden dimension has significant effect on the model performance

4.1.2 Variation across number of layers layers

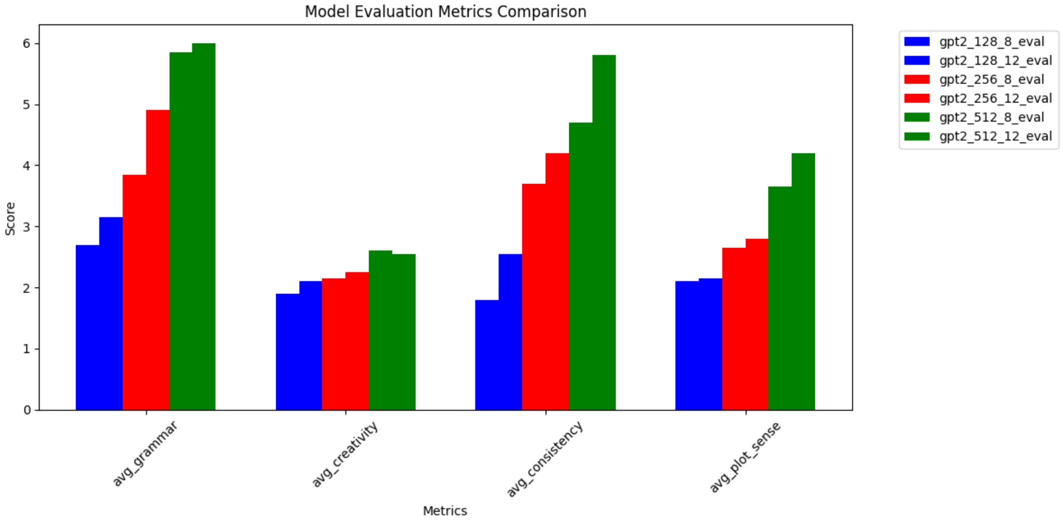


Figure 2: We can see that number of layers also has a significant effect on the model performance

4.1.3 Variation across number of heads

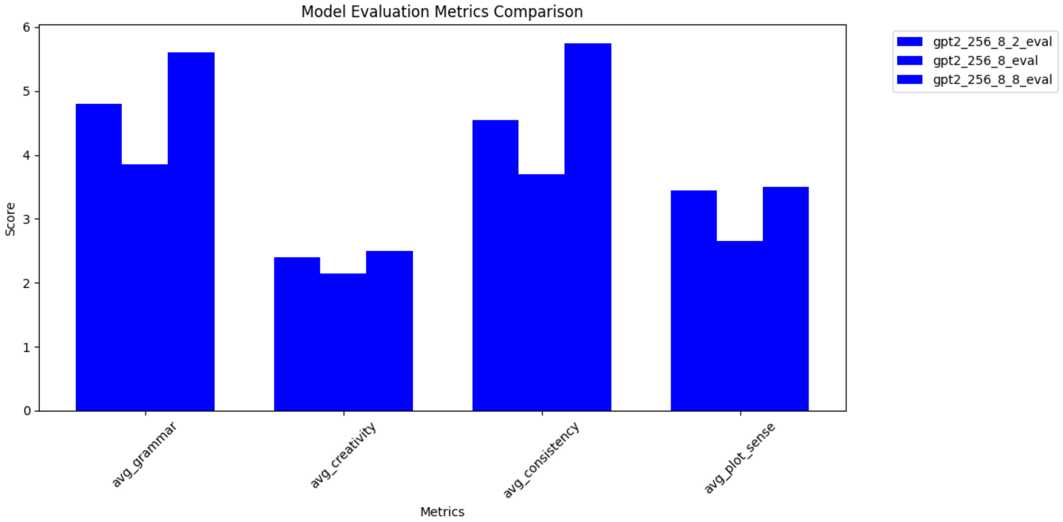


Figure 3: We can see that number of heads doesn't have a significant effect on the model performance

4.1.4 Comparison with pretrained LLMs

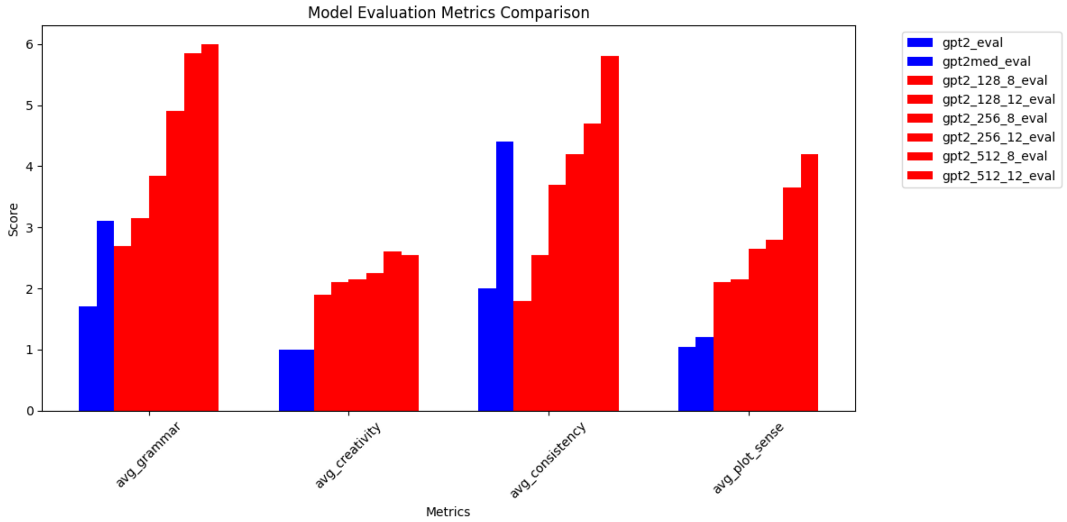


Figure 4: We can see that despite having fraction of parameters, SLMs surpass LLMs in short story generation

4.2 Knowledge, Reasoning and Context Tracking

Prompt Types:

- **Factual Prompts:** These test the models' knowledge of common sense facts.
- **Reasoning Prompts:** These assess basic reasoning abilities, such as understanding cause and effect or making eliminations.
- **Consistency (Context-Tracking) Prompts:** These evaluate the models' ability to maintain coherence and continuity with the given context, including character names, actions, settings, and plot details.

Message	**256-8-4**	**512-8-4**	**256-8-8**	**512-12-4**	**contrast-256**	**contrast-512**
`Both Ben and Lily wanted cake. Father said that there was only one piece of cake left. They`	said that they were good for their birthday.	said that they should share and be kind.	said they could have a cake.	said they had to wait until the cake was ready.	said it was a cake for them.	said that they should not have to share.
`On weekends Jack went to visit his grandmother whereas on weekdays he would go to school. Last weekend, when Jack was on his way to`	school, he was very excited to see all the different things he had.	school, he saw a big blackboard.	school, he saw a big, scary monster.	school, he was excited to see all the new things he could do.	school, he saw a big, red car.	school, he saw his friend Sam.
`Jack's mother was not home, and his father was at home. When Jack came home, he said hello to`	his parents	his father.	his mother and said, Ï'm here to visit me.	his mother.	his father.	his father.

Table 2: Reasoning Evals

Message	**256-8-4**	**512-8-4**	**256-8-8**	**512-12-4**	**contrast-256**	**contrast-512**
`Alice was so tired when she got back home so she went`	to bed	to bed.	to bed.	to the kitchen to get some food.	to the park.	to the hospital
`Jack wanted to read a book, so he went to`	the park.	the store.	the park	the library	the store	the library
`Jack and Lily saw a rainbow after a rainy day. They were amazed by the colors. Jack said, "Look, Lily. A rainbow has`	a big yellow flower	a rainbow in the sky	many colors and shapes.	a rainbow.	a rainbow in the sky	many colors

Table 3: Factual Evals

Message	256-8-4	512-8-4	256-8-8	512-12-4	contrast-256	contrast-512
`Hi Jane, have you seen Alice? I can’t find her anywhere", said Jack.`	Jane was very excited.	I can’t believe it.	Jack said, ”Let’s go to the park.”	She was very happy to see her.	Jane said, ”I’m sorry, I will find you.	Jane was very excited.
`Max had two dogs. One was white and the other was black. Max walked up the street and saw a kid with a dog. He told the kid, "I see you have a Brown dog. I also have`	a dog and a dog.	a dog.	a big dog.	a dog.	a dog!” Max was very happy.	a dog.
`Anne had a piece of candy in her left pocket and a piece of chocolate in her right pocket. Anne’s mom asked her, "Anne, what is that you have in your left pocket?"`	it is a special treat	a candy.	I’m sorry, I just wanted to share my candy with you.	it’s a special piece of candy.	Mom said, that’s a great idea, I will take a bite of your candy	please have some candy in my pocket.”

Table 4: Context Tracking Evals

4.3 Quantitative measurement of similarity using Rouge score

4.3.1 Similarity between Generations and Original stories

The histograms below show the ROUGE-2 score evaluation for the different models between stories generated by the models with the original story from the dataset. Seed stories are generated by taking some random stories from the train split of the dataset and taking only about first 40% of it and asking the model to fill in the rest to give the generated story. This analysis confirms that the small language models are actually learning the language and merely not memorizing the stories from the training set.

This is an important evaluation to do to confirm generalizing capability of the trained models and have further deeper insights into their working.

4.3.2 Similarity across generations

In the below graphs we calculate the max rouge scores across 100 generations. The result is plotted in the histogram below. The low rouge score indicates that the generations are diverse and depend on the seed text and are not repeated across generations.

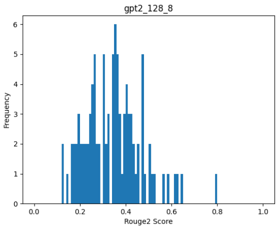


Figure 5: ROUGE-2 Score for 128_8

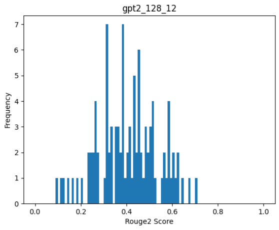


Figure 6: ROUGE-2 Score for 128_12

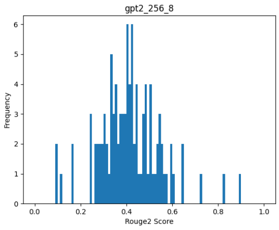


Figure 7: ROUGE-2 Score for 256_8

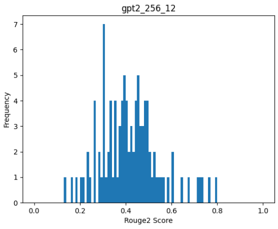


Figure 8: ROUGE-2 Score for 256_12

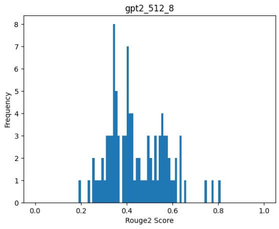


Figure 9: ROUGE-2 Score for 512_8

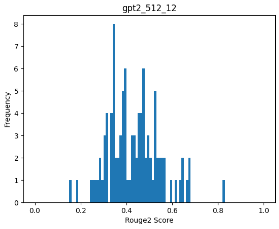


Figure 10: ROUGE-2 Score for 512_12

Figure 11: ROUGE-2 Score Evaluation for Different Models

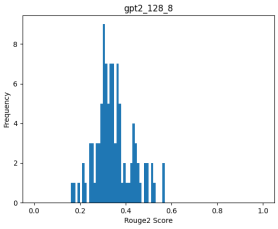


Figure 12: ROUGE-2 Score for 128_8

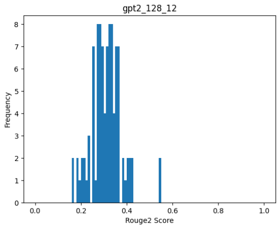


Figure 13: ROUGE-2 Score for 128_12

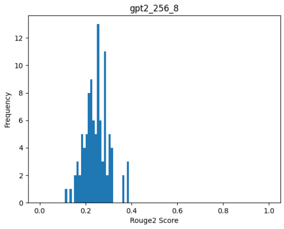


Figure 14: ROUGE-2 Score for 256_8

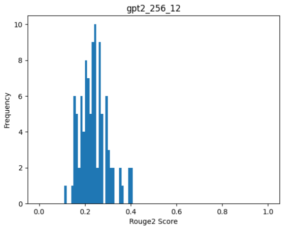


Figure 15: ROUGE-2 Score for 256_12

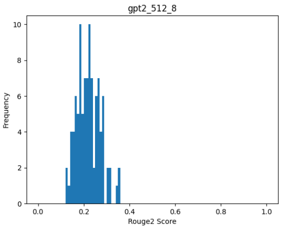


Figure 16: ROUGE-2 Score for 512_8

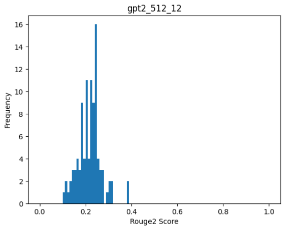


Figure 17: ROUGE-2 Score for 512_12

Figure 18: ROUGE-2 Score Evaluation among generations for Different Models

4.4 Instruction following Capabilities

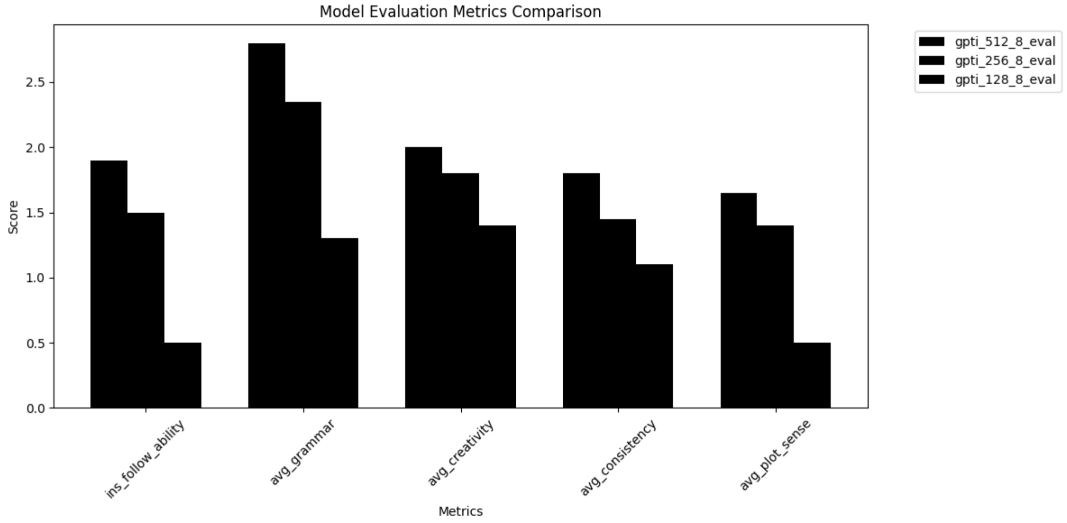


Figure 19: Comparison of models trained on TinyStoriesInstruct

Summary: A little bird falls from a branch in the park and is rescued by a kind girl who takes it to a safe place. The bird sings a happy song for the girl afterwards.
Words: shout, branch, loud
Story:
Once upon a time, there was a little bird. The bird was so excited to go and sing a song. One day, the bird was feeling a bit too scared to move. It was a **loud** noise, but the bird was brave and flew over to the **branch**.
The **branch** was scared and started to cry. It was a kind voice, "Don't cry, little bird. I will help you. I will take you to a safe place."
The bird was so happy and thanked the little girl. She flew away and sang a happy song for the bird.

Example generation by instruct model of hidden dimension 512. Everything after "Story:" is generated by the model

We train the transformer model from scratch on TinyStoriesInstruct dataset. The dataset has Words, Summaries and Features of a Story along with the story itself. We see that even though Gemini grades the instruction following capabilites of the trained model poorly, manual evaluation suggests some basic understanding of the instruction.

4.5 Conclusions

- The models trained from scratch give better stories than pretrained Models like gpt2 with only a fraction of parameters.

- By just learning on the stories the transformer models do pretty well on factual, context-tracking and resoning tasks.

- The models actually learn the strructure of grammar and English Language and do not just recite the stories it has seen.

- Transformer models are also capable of following instructions to some extent if trained on proper data though this does hinder other abilities such as their story-telling abilities as seen in the instruct models.

- We also see that Gemini being less capable than GPT-4 does harsher evals for all the stories.

- We can conclude as well that the evaluation proposed in the paper is not a good way to evaluate models. It heavily depends on LLM being used and is not reliable.

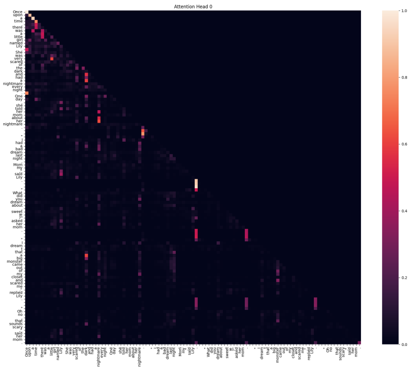
5 Interpretability

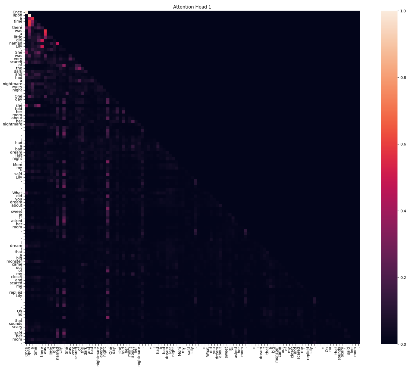
Components of a neural network may not have distinct roles and may behave in a complex and messy way. In order to better understand their working, it becomes important to delve into the intricacies of these components to see what exactly is happening. The paper presents some preliminary evidence that training smaller models on TinyStories leads to higher interpretability mainly because of their small size. We particulary focus on the attention heads of the model and the neurons in the MLP. Attention heads exhibit diverse and meaningful functions, such as attending to the previous word, the subject of the sentence, the end of the sentence, or the main topic of the story. We identify the most influential tokens in the MLP for each neuron. Some neurons are found to get activated on words that have a specific role in the sentence.

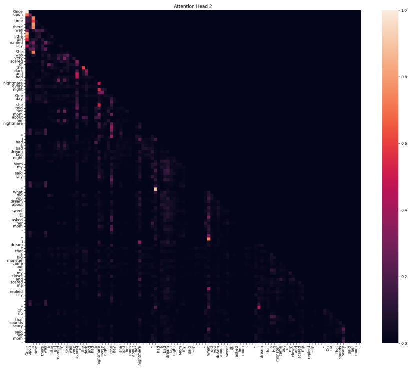
5.1 Interpretating Attention Heads

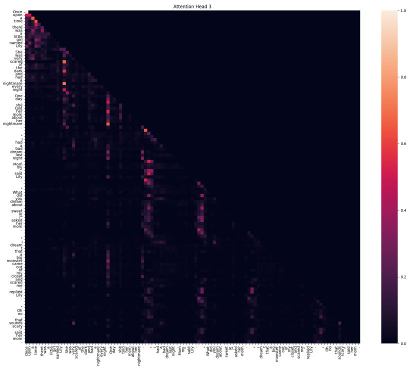
We fetch the attention patterns from the model output on a particular piece of input. For every head, we generate a heat map from its attention pattern. We then examine the correlation between the token distances and the attention weights. If the correlation is greater than 0.5, we conclude that the attention is postional, else it is semantic. [1]The heat maps for `gpt2_512_12` model are as:

[1] Heat maps for other models can be found in the folder submitted







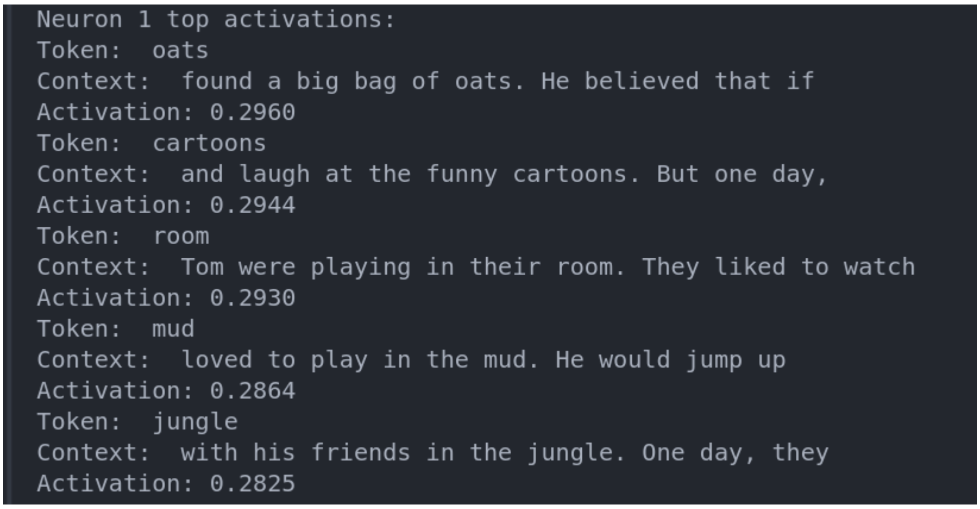


We see that the correlation is very less for all the four heads which implies that all the four heads are involved in providing semantic attention.

5.2 Interpreting the roles of neurons

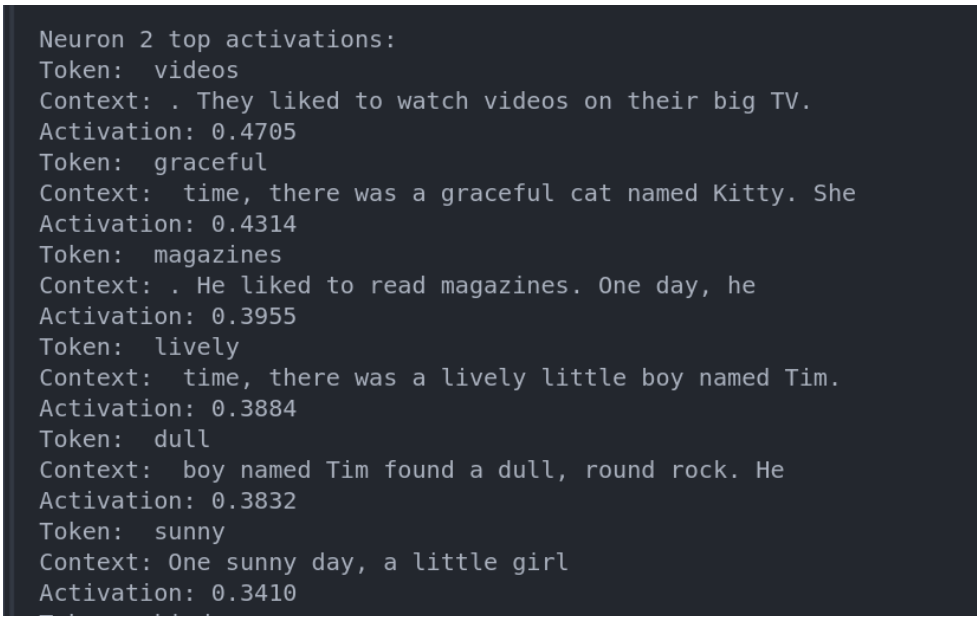
To do this, we fetch hidden states from the model outputs and then use them to track tokens with the highest activation for each neuron. Below we show results for the `gpt2_512_12` model after analyzing neurons in some of its layers.

We observe that certain neurons in certain layers show higher activations for particular kinds of words. For example as we show below, the neuron 1 in layer 4 shows higher activations for nouns as compared to verbs, prepositions, adjectives, etc. while the neuron 2 in layer 1 shows higher activations for adjectives as compared to other kinds of words. This shows that the network does get some sense of how the language is constructed.



```
Neuron 1 top activations:
Token:  oats
Context:  found a big bag of oats. He believed that if
Activation: 0.2960
Token:  cartoons
Context:  and laugh at the funny cartoons. But one day,
Activation: 0.2944
Token:  room
Context:  Tom were playing in their room. They liked to watch
Activation: 0.2930
Token:  mud
Context:  loved to play in the mud. He would jump up
Activation: 0.2864
Token:  jungle
Context:  with his friends in the jungle. One day, they
Activation: 0.2825
```

Figure 20: Neuron 1 of layer 4 having highest activations on nouns



```
Neuron 2 top activations:
Token:  videos
Context: . They liked to watch videos on their big TV.
Activation: 0.4705
Token:  graceful
Context:  time, there was a graceful cat named Kitty. She
Activation: 0.4314
Token:  magazines
Context: . He liked to read magazines. One day, he
Activation: 0.3955
Token:  lively
Context:  time, there was a lively little boy named Tim.
Activation: 0.3884
Token:  dull
Context:  boy named Tim found a dull, round rock. He
Activation: 0.3832
Token:  sunny
Context: One sunny day, a little girl
Activation: 0.3410
```

Figure 21: Neuron 2 of layer 1 having high activations on adjectives

6 Contrastive Training

From the above results we see that grammar is learnt much better than creativity and plot sense. Hence we introduce a new way to train the LM in order to improve these metrics on the same dataset.

6.1 Method

During each train loop, we take 33% of the stories and shuffle the sentences in these stories. Along with the Cross Entropy Loss that is normally calculated, $\mathcal{L}_n$, We calculate Contrastive Loss, $\mathcal{L}_c$ for the next word prediction of the story with shuffled sentences.

$$\text{We calculate the overall loss as: } \mathcal{L} = \mathcal{L}_n + \frac{\alpha}{\mathcal{L}_c + 0.01}$$

Hence we try to 'maximize' the loss on the shuffled stories. α is taken as a hyperparameter. We take $\alpha = 5$ for the trainings.
0.01 in the denominator prevents exploding gradient.

6.2 Results

Even though the shuffled stories have proper grammar, We see that the grammar score is not impacted significantly as it learns grammar from the rest of the data. We can see improvements across all other parameters. In all parameters other than grammar, the model with hidden dimension 256 but trained with the contrastive loss method performs on par with the model of dimension 512 trained normally!

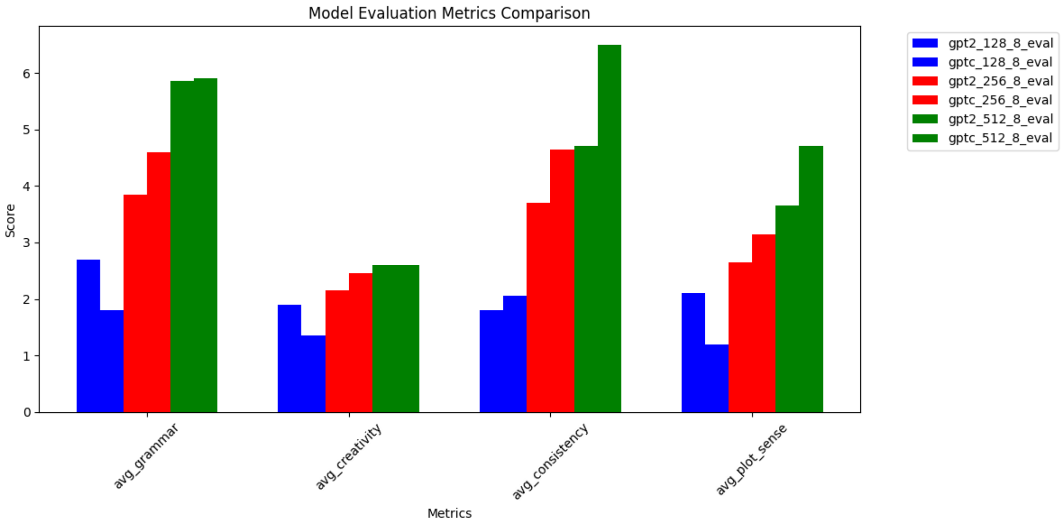


Figure 22: Contrastive Model Evaluations

7 Improving Memory Efficiency

As the model seems to take a large amount of GPU memory we also tried to improve this memory footprint. A large amount of memory taken by the transformer is a result of the n^2 attention matrix that is constructed during the forward step. To resolve this we use a Encoder only model that takes in tokens

2 at a time, and gives out 1 token, discarding some information but hopefully learning to hold on to the important info. These are then sent to the decoder model that computes the next token in the sequence. These tokens are sent as a source to the decoder model which takes the input sequence Tensor as the target and attempts to get the next two tokens decoded.

7.1 Architecture

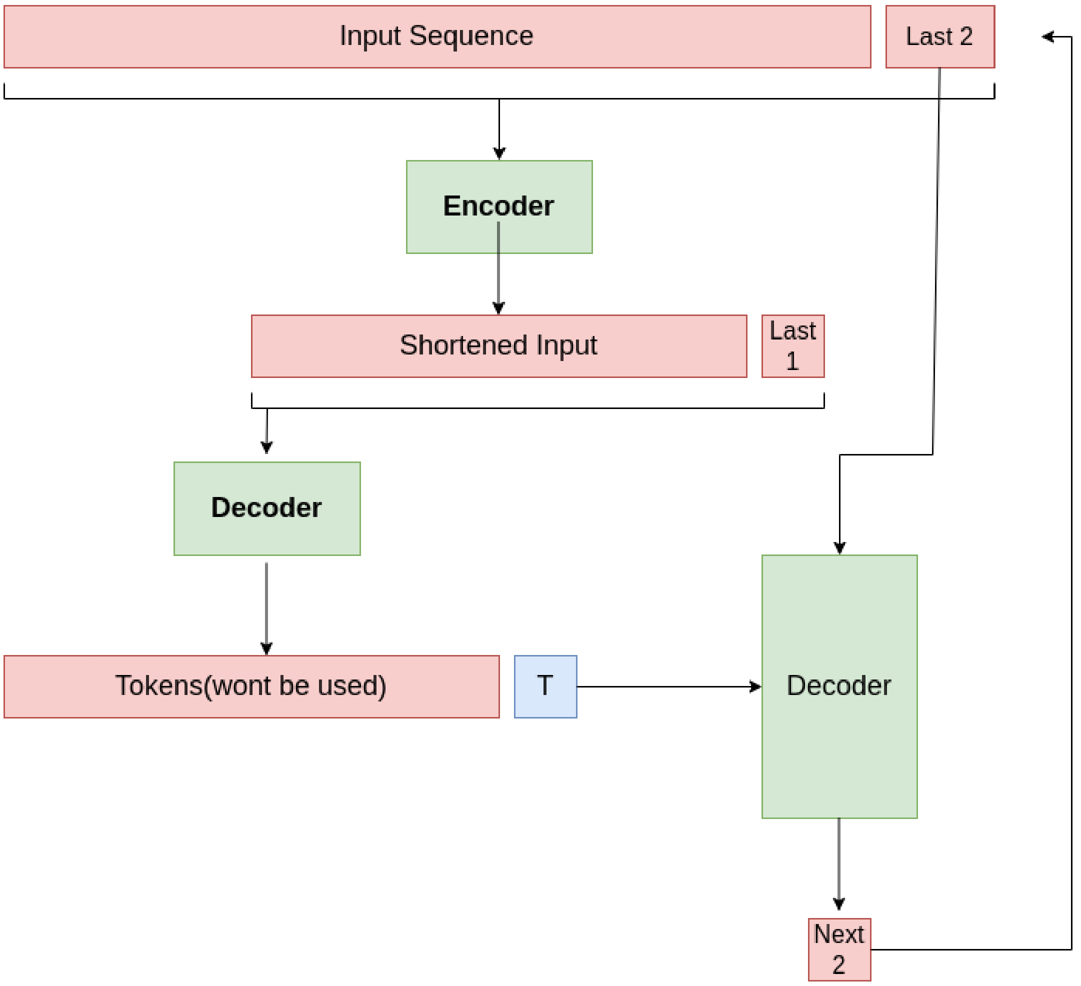


Figure 23: Architecture used for memory efficiency

7.2 Memory Evaluation

Table 5: Memory comparison for GPT2 and Compressor across batch sizes. All sizes are in MB

	Batch Size 2	Batch Size 4	Batch Size 8	Batch Size 16	Batch Size 32
GPT2					
128	904	2465	4845	9606	oom
256	1076	2930	5691	oom	oom
512	1448	3933	7463	oom	oom
Compressor					
128	457	1272	2361	4517	8831
256	720	1936	3451	6471	oom
512	1351	3408	5781	10529	oom

7.3 Model Eval

Table 6: Evaluation of models across different metrics

Model	Grammar	Creativity	Consistency	Plot Sense
comp-128-8	1.85	1.9	1.5	1.3
comp-256-8	2.6	2.1	2.02	1.6
comp-512-8	3.5	2.21	2.6	2.25

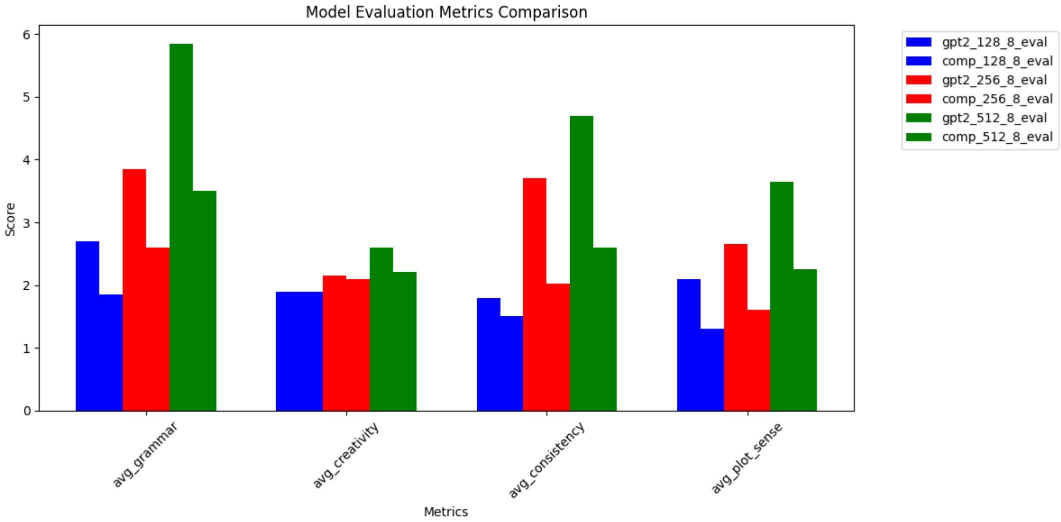


Figure 24: Comparison between Compression and normal model

8 Loss Curves

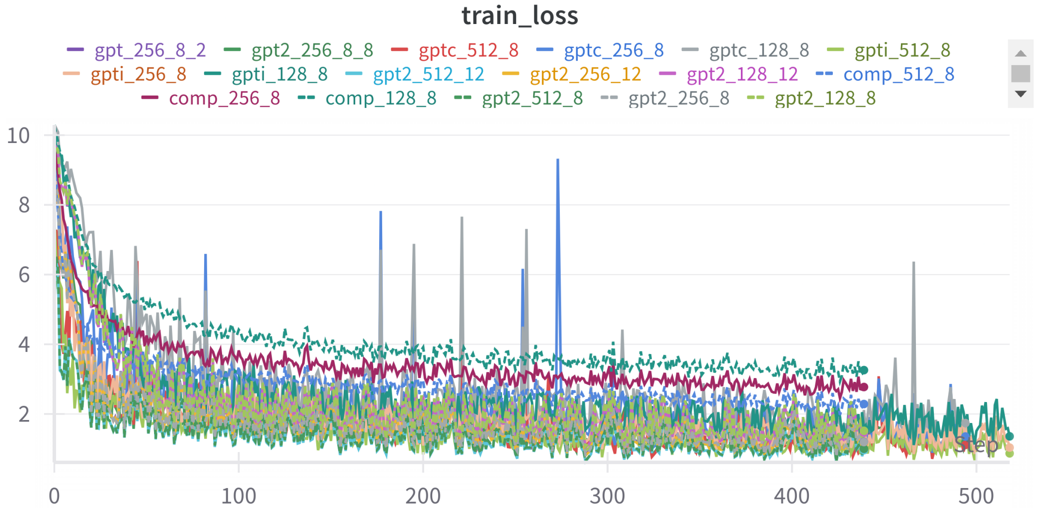


Figure 25: Loss evolution on train set

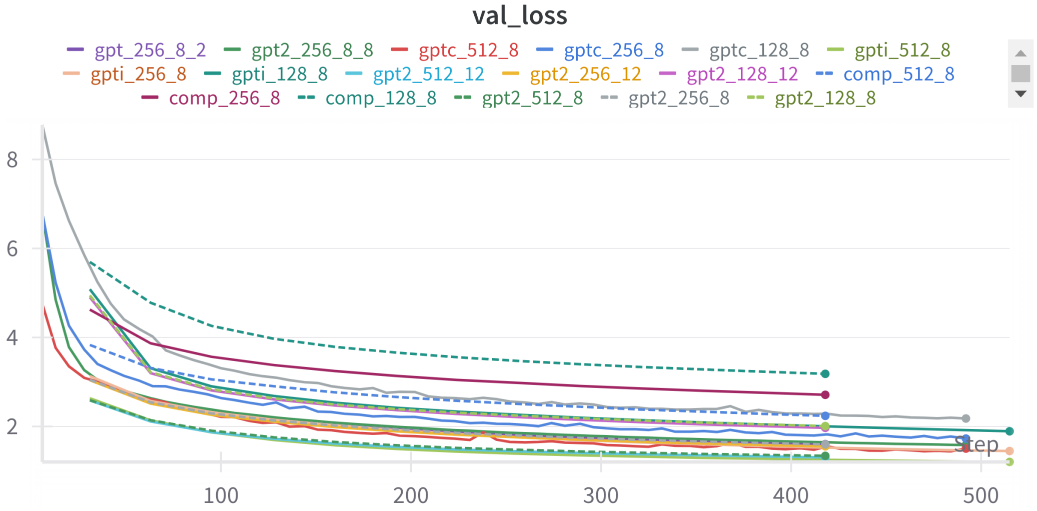


Figure 26: Loss evolution on validation set

We can see some spikes in Loss in Contrastive model training due mostly due to the denominator getting too small. Otherwise, the loss curve is very smooth. We have not trained the model fully due to resource constraints. The model is expected to perform much better if trained for longer.

We can also see that Compression model gets trained the slowest and achieves maximal loss among all the models.