# A Bedtime Story Generator Using Emojis by Fine-tuning LLM


Yu-Chien Lin
CS
109062139
*liniris2001@gmail.com*

Desmond Cheong Zheng Yong
*EECS*
109006281
dzycheong@gmail.com

Yu Ting Jen
CS
109000125
*jennyyu154@gmail.com*

Chou Kuan Lin
EECS
110060011
fizz92119@gmail.com

Yang Wen Hsuan
MATH
*109021221*
cosmo20020702@gmail.com



***Abstract*— This paper explores the utilization of Llama 2[1], a fine-tuned Large Language Model (LLM), for generating bedtime stories using emoji-based language. By incorporating LoRA (Low-Rank Adaptation), the study showcases a novel approach to NLP tasks, optimizing the balance between performance and computational efficiency. Results indicate that Llama 2, integrated with LoRA, can produce engaging, diverse content with a significantly reduced parameter footprint, broadening the potential for creative applications in NLP.**



***Keywords*— Llama 2, LoRA, NLP, Emoji Language, Bedtime Stories, Fine-tuning, Large Language Models, Computational Efficiency, Story Generation.***


## I. Introduction

In the rapidly advancing field of machine learning, the use of emojis in natural language processing offers a simple yet powerful way to communicate emotions and ideas. Their straightforward nature and vast combinatory possibilities make them an ideal tool for innovative applications like bedtime story generation. This project leverages the capabilities of a Fine-tuned Large Language Model (LLM) to craft engaging narratives using emojis, pushing the boundaries of conventional storytelling. Recognizing the need for diverse and captivating content, this approach not only appeals to a broad audience but also aims to enchant children with delightful bedtime stories. We chose emojis for our project because they are simple to use, can powerfully express ideas, and can be mixed and matched in lots of ways. With the fine-tuning we have applied, our model is now capable of generating complete bedtime stories from just three emojis, without needing a full sentence as a prompt. The illustration of our model is shown in Figure 1. This introduction paves the way for an exploration of the methodologies, datasets, and compelling outcomes of this unique venture.

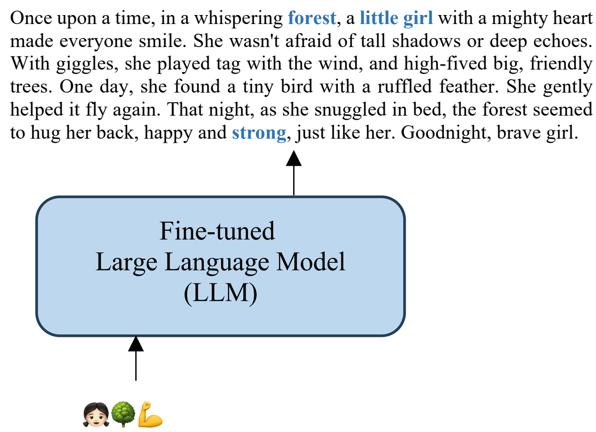



Figure 1: Example input and output of our model

## II. Methods

### *A. Llama 2 pretrained weigth*

Llama 2 is an advanced open-source large language model (LLM) with versions ranging from 7 billion to 70 billion parameters. Released by Meta, these models have been improved over their predecessors, offering a longer context length and efficient inference capabilities. The fine-tuned models, especially Llama 2-Chat, are optimized for dialogue applications and demonstrate performance on par with other prominent models like ChatGPT in benchmarks. Llama 2 and its models are integrated into the Hugging Face ecosystem, facilitating easy access and deployment for various applications. This release also features a permissive community license, making it available for commercial use.

### *B.* LoRA (Low-Rank Adaptation)

LoRA[2] is a method of adapting large pre-trained models like Llama 2 for specific tasks while maintaining parameter efficiency. LoRA works by freezing the original model weights and introducing low-rank matrices that are trainable. This approach modifies the weight matrices in the Transformer layers of the model during adaptation. By using rank decomposition, LoRA reduces the number of parameters that need to be trained, thereby reducing memory requirements and computational cost. It avoids retraining the entire model, making it efficient for deployment, especially when dealing with very large models.

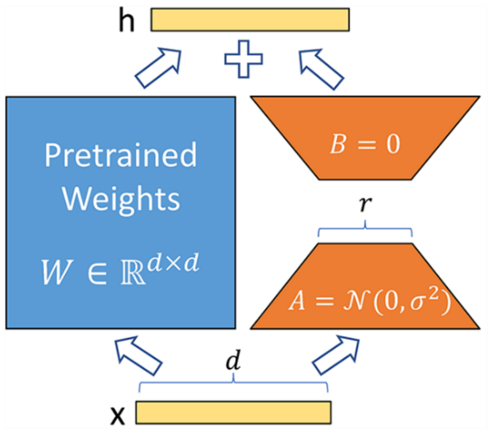
### *C.* Mathematical Aspects of LoRA

The mathematical construction of LoRA (Low-Rank Adaptation) involves introducing low-rank matrices $A$ and $B$ to modify the weight matrices W of a neural network's Transformer layers during fine-tuning. This is achieved by freezing the original pre-trained weights $W_0$ and updating them with the product of $A$ and $B$, effectively representing the change in weights $\Delta W$ as $BA$, where $B$ is a matrix of size d×r and $A$ is a matrix of size $r\times k$, with $r$ being much smaller than $d$ or $k$. The rank r represents the "intrinsic rank" or the effective degrees of freedom in the adaptations made to the original model weights.

In practice, during the forward pass of the model, instead of using the full weight matrix $W$, the modified weights $W_0$ + BA are used. This update is low-rank parametrized, which means that the trainable parameters are significantly fewer than in the original model, making the fine-tuning process more efficient in terms of computational resources.

Mathematically, the forward pass for an input using a weight matrix $W$ can be modified with LoRA as follows:

$$h = W_0\mathbf{x} + \Delta W\mathbf{x} = W_0\mathbf{x} + \mathbf{BAx}$$




Fisure 2: Illustration of LoRA

### D. Enhancing Efficiency in Large Language Model Fine-Tuning with LoRA

LoRA offers significant advantages for fine-tuning LLMs like Llama 2. It allows for parameter-efficient adaptation by inserting low-rank trainable matrices alongside frozen pre-trained weights in neural networks. This approach significantly reduces the number of trainable parameters, sometimes by factors of up to 10,000 times, and reduces memory requirements during training. LoRA can perform comparably or even better than full model fine-tuning while offering higher training throughput and avoiding additional inference latency commonly seen with other adaptation methods. Additionally, LoRA can be combined with other tuning methods like prefix-tuning for further efficiency. This makes LoRA a powerful tool for adapting LLMs for specific tasks without the usual computational and storage overheads.

### E. Fine tuning Llama 2

Llama 2 is fine-tuned by installing the required libraries and loading the pre-trained model and tokenizer. The model is then adapted using parameter-efficient techniques like LoRA, which is crucial for memory management. Additionally, techniques like 4-bit quantization optimize memory usage. Training parameters are configured, followed by the model's fine-tuning with a specific dataset using a specialized training system. Lastly, the model's performance is evaluated.

## III. Data and Code Avalibility

### A. Dataset

The training dataset that we use consists of 785 instances, each of them has three emojis as the input of our model, which representing the main role, the location, and the purpose of the story, and a bedtime story as the expected output. Because of the lack of this kind of dataset on open resources, we create the dataset by ourselves using ChatGPT as a tool. We choose some emojis and use their combinations as the limitation of our prompts to ChatGPT to generate a story. Each combination will be used in different prompts several times to get various stories. Then we manually inspect the stories that are not fluent, and re-prompt the emojis to get a fluent story. By this, we construct a reliable dataset to train our model.

The test set comprises 12 different emoji combinations, and for each combination, we have prepared five different reference stories. Multiple reference stories allows for a more flexible BLEU score evaluation, providing a better understanding of our model's ability to generate creative stories.

### B. Code availabilty

We push the dataset we construct to Hugging-Face, hoping that the dataset can help others doing similar researches. Here is the link to our dataset on Hugging-Face: “https://huggingface.co/datasets/lemonpuddi/ml-emoji-story”

### C. Code availabilty

We also maintain a repository on Github to trace our work and share our result with others. Here is the link: “ https://github.com/lyc2001/bedtime_story_generator”

## IV. Results

### A. Generated Stories Examples

In Table 1, we present two instances of story generation by our fine-tuned model. The first entry corresponds to the input emojis 🦔🏰📚, which symbolize a hedgehog, a castle, and books, respectively. The role, a hedgehog, is naturally integrated into the setting of a castle, engaging in activities associated with learning and reading, as denoted by the books’ emoji. Notably, the generated text contains words such as 'castle,' 'learning,' 'library,' and 'hedgehog,' which are directly related to the input emojis, underscoring the model's capacity for contextually aware generation. The story is not only relevant to the given input emojis but also maintains a coherent structure, making it a successful outcome for our bedtime story generator.

In the second example showed in Table 1, the input emojis are 👧🏫👫, representing a girl, a school, and friendship. The narrative constructed by the model revolves around Lily, a girl whose experience with a pen pal, Emma, blossoms into a cherished friendship. Key terms like

| Input | Output |
|---|---|
| 🦔🏰📚 | In a grand castle, there lived a little hedgehog named Henrietta. She loved learning and found a magical library within the castle walls. Every day, Henrietta explored the shelves, reading stories of brave knights and wise dragons. She became the smartest hedgehog in the castle, teaching her friends all she had learned. The castle's people celebrated Henrietta's love for books, making her the cleverest hedgehog in the kingdom. She knew that knowledge was power, and with every page she read, she grew smarter and smarter, becoming the wisest hedgehog in the entire land. Henrietta lived happily ever after in her magical castle, surrounded by the wonders of reading and learning. The end. |
| 👧🏻🏫👭 | In a quiet school, there was a girl named Lily. One day, she received a surprise letter in her lunchbox. It was from her pen pal, Emma, who lived on the other side of the world. Lily and Emma started exchanging letters, sharing stories and dreams. Lily's teacher, Mrs. Smith, noticed the special bond between them. She organized a pen pal day at school, where Lily and Emma met for the first time. They hugged each other tightly, feeling like old friends. Lily realized that even though they were far apart, friendship could cross oceans. That day, the school became a little brighter, thanks to Lily and Emma's endless letters and laughter. As the sun set, Lily knew that she had found a true friend in Emma, and their bond would last a lifetime. |

Table 1: Examples of the stories generated by our model. Highlighted terms are highly related to input emojis

| Method | BLEU |
|---|---|
| Baseline (Llama 2) | 2.11 |
| ChatGPT | 1.91 |
| Bedtime Story Generator (our model) | **8.10** |

Table 2: BLEU score evaluation results

| Method | GPT Grader (Baseline vs our model) | GPT Grader (ChatGPT vs our model) |
|---|---|---|
| Baseline (Llama2) | 0.08 | - |
| ChatGPT | - | 0.25 |
| Bedtime Story Generator (our model) | **0.92** | **0.75** |

Table 3: Results of GPT evaluation

'school,' 'girl,' 'teacher,' and 'friends' match the emojis given and are used throughout the story, showcasing the model's skill in producing content that is both related and engaging. The emergence of themes such as friendship crossing distances and the enriching nature of shared stories accentuates the model's sophisticated understanding of the prompts.

After fine-tuning with our dataset, the output from the model is evidently suitable for children. The language used is straightforward, employing vocabulary and grammar that are easy for kids to understand. Moreover, the content is not only age-appropriate but also conveys messages that are beneficial for children, such as the importance of reading, learning, and friendship. This demonstrates the fine-tuning process's success in adapting the model's outputs to be both enjoyable and instructive for kids, ensuring that the generated stories serve as both entertaining and educational resources for children.

### B. *BLEU Score Evaluation*

Traditionally, BLEU Score is predominantly utilized for assessing translation tasks by measuring the correspondence between a machine's output and that of a human. Given that the task of generating bedtime stories does not have a definitive or widely-recognized numeric evaluation method, due to the subjective nature of creative outputs, we sought to adapt the BLEU score for our purposes. Creativity and imagination play pivotal roles in storytelling, which typically resists quantitative assessment. However, considering our model interprets emojis as a form of language that it 'translates' into narrative text, the BLEU score provides a means to quantitatively evaluate our model's performance.

Additionally, one of the advantages of the BLEU score is its capability to consider multiple reference texts simultaneously, which is particularly beneficial for tasks where there can be many correct ways to express the same idea. In our evaluation, to accommodate the inherent diversity of creative storytelling, we provided five reference stories for each emoji combination in our test set. This approach allows for a more flexible and forgiving assessment, recognizing the varied yet valid ways in which a story can unfold from the same set of emojis. This makes BLEU a more suitable metric for our task, where translating a set of emojis into text can yield numerous appropriate and imaginative stories.

As Table 2 illustrates, our model significantly outperforms the baseline Llama 2 model and the ChatGPT according to this metric. This suggests that, despite the inherent limitations of using BLEU for a creative task, it can offer insights into the cohesiveness and relevance of the generated stories in relation to the symbolic input provided by emojis. It is important to note that while BLEU scores can provide an objective measure of textual similarity, they do not capture the full essence of narrative creativity or emotional resonance, which are integral to bedtime stories. Therefore, these BLEU score results should only be interpreted as one part of a broader evaluation framework; a story that meets the approval of our evaluators is ultimately considered a success.

### C. GPT Evaluation

In addition to the BLEU score evaluation, we utilized GPT as an evaluative tool, asking it to compare and rate the stories produced by both Llama2 and our bedtime story generator model. As shown in Table 3, when comparing the output of Llama2 against our model, 92% of the time stories generated by our model were preferred. Similarly, when the output of ChatGPT was compared to our model, our model's stories were favored 75% of the time. This comparative evaluation further demonstrates our model's effectiveness in creating engaging bedtime stories.

$$\text{Model Score} = \frac{\sum \text{Better Stories}}{\sum \text{Test Cases}}$$

## V. Conclusions and Discussions

### A. Conclusions

Our project has successfully demonstrated the capability of a fine-tuned Llama 2 model, now optimized with Low-Rank Adaptation (LoRA), to generate simple bedtime stories from just three emojis. By adopting LoRA, we have significantly enhanced the model's efficiency, reducing the resources required for operation. This advancement allows our model to interpret and creatively transform a minimal input of three emojis into enchanting bedtime narratives, showcasing both the model's effectiveness in NLP tasks and its potential in resource-conscious applications

### B. Discussions

The dataset's role in model training is pivotal, especially when considering the variance between new and original datasets in fine-tuning. Our project used Llama 2, which inherently can generate stories. Our new dataset forms a bridge between three emojis and these narratives. Remarkably, this close alignment between the new and original datasets contributed to efficient fine-tuning, despite our dataset's modest size of only a few hundred entries. However, we encountered an issue with the model generating excessively lengthy and repetitive responses, up to 500 words. This issue persisted despite dataset checks and hyperparameter adjustments, suggesting a potential bug in the fine-tuning code, a suspicion supported by similar experiences within the community.

## Author Contribution Statements

The TA will report our contribution based on the anonymous survey.

## References


[1] TOUVRON, Hugo, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[2] HU, Edward J., et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[3] https://www.datacamp.com/tutorial/fine-tuning-llama-2

[4] https://chat.openai.com/