SI 630 WN24 Project Update

Short Story Ending Generation

Divya Santhanam and **Divyam Sharma**

1 Introduction

Writing stories is an engaging yet challenging endeavor. Often, authors encounter moments of creative block, where the path forward in their narrative becomes obscured. This project is designed to address such moments by providing an innovative solution: a tool that completes stories based on given prompts. By inputting a short story prompt, users can receive a conclusion to their story, articulated in one sentence or more, thereby enhancing the storytelling process with AI-driven creativity. This tool aims not only to assist authors in navigating writer's block but also to offer a fun and interactive way for anyone to expand on story ideas spontaneously. Through this project, we explore the intersection of artificial intelligence and creative writing, pushing the boundaries of how stories can be crafted and concluded.

As discussed earlier, our project aims to develop a sophisticated text-generator that effectively writes an applicable and readable conclusion to a short story prompt. Our text-generator will require four sentences of text input that represent the bulk of the story. From this text, our generator will produce a one-sentence-long conclusion. The generator will be trained on thousands of five-sentence-long stories that resemble the type of user input that we would expect to see.

Additionally, once our model has been trained and performs well as determined by several readability metrics, we plan to conduct additional training on 30-40 longer stories to see how our model performs on generating longer conclusions.

Our project aims to solve the problem of writer's block. There are several instances where having a tool that can conclude a story would be beneficial. For example, a potential user of our text-generator could be an author looking for a conclusion to their short story, or a parent that is looking for a personalized bedtime story to tell their child.

2 Data

The data we will be using for our project will be five-sentence-long short stories. To obtain the short stories, we have accessed a ROCStories Corpora database maintained at Rochester University.

The dataset we plan to use is from the Story Cloze Test and ROCStories Corpora which has 98,161 short five-sentence-long stories that can be found here. The dataset has free access to anyone upon request, which we have already received. This project aims to evaluate story prediction. Given two possible conclusions to a story, the model chooses the more favorable option. Our project extends this concept by focusing on text generation over prediction. Rather than being given options to choose from, our model will generate the text itself.

The ROCStories Corpora consists of data obtained during the spring of 2016 and the winter of 2017. The quantity of observations from both datasets can be seen in Table 1.

ROCStories 2016	ROCStories 2017
45496	52665

Table 1: Quantity of observations in ROCStories Corpora

For the purposes of this project, we will treat the two datasets as one. The datasets consist of five sentences for each observation. In building our final model, after training, we will use four sentences as input and evaluate the output against the actual fifth sentence in the story.

An example observation from our dataset is shown in Table 2.

Category	Data
ID	9a51198e...
Title	Overweight Kid
S1	Dan's parents were overweight.
S2	Dan was overweight as well.
S3	The doctors told his parents...
S4	His parents understood and...
S5	They got themselves and...

Table 2: An example observation from the ROCStories dataset.

3 Related Work

A similar work that has been done by (Mostafazadeh et al., 2016) at the University of Rochester has focused on coming up with a benchmark - Story Cloze benchmark as a method for evaluating the understanding of short (five-sentence) commonsense stories. In the Story Cloze benchmark, the first four sentences of an everyday story are given, and the concluding sentence is withheld. The model is offered two choices, both of which are contextual to the preceding sentences but one of which is right and the other is wrong. So the focus of the above-mentioned work is on predicting the right ending out of the two options. Other works on similar lines focusing on predicting the right ending to solve ROCStory Cloze Task have been such as (Schwartz et al., 2017). To improve the performance, features like topic words and sentiment score are also extracted and incorporated (Chaturvedi et al., 2017). Neural network models have also been applied to this task (e.g., (Huang et al., 2013) and (Cai et al., 2017)), which use LSTM to encode different parts of the story and calculate their similarities. In addition, (Li et al., 2018) introduces event frame to their model and leverages five different embeddings. Finally, (Radford et al., 2018) develops a transformer model and achieves state-of-the-art performance on ROCStories, where the transformer was pre-trained on BooksCorpus (a large unlabeled corpus) and finetuned on ROCStories.

Our intended work will focus on being able to generate the whole last line of the 5-sentence story, aiming to have contextual meaning and logical coherence with the rest of the story (the 4 seen sentences) and not just selecting the right ending choice out of a given few options.

4 Methodology

4.1 Preprocessing Steps:

- **Tokenization:** Split the text into tokens (words or subwords) using a tokenizer suited for the chosen model architecture (e.g., BPE tokenizer for transformer-based models).
- **Cleaning:** Remove or correct typographical errors, standardize quotation marks, and handle special characters to ensure text consistency.
- **Segmentation:** Divide each story into two parts: the body (beginning and middle) and the ending. This segmentation facilitates the model to learn the transition from the story body to its ending.
- **Vectorization:** Convert tokens into numerical vectors using the tokenizer's vocabulary. This step is crucial for feeding textual data into neural networks.

4.2 Model Selection and Architecture

4.2.1 Model 1: Large Language Model

- **Pre-trained Language Model:** Will load a transformer-based pre-trained language model like GPT-2 or DistilGPT-2 or GPT-3.5 now as credits have been offered for it, because of their proven capacity for generating coherent text. The choice of model size (among GPT-2, GPT-3.5 and DistilGPT-2) should balance between computational resources and desired output quality.
- **Fine-tuning:** Adjust the model to the task of short story ending generation by fine-tuning it on the preprocessed dataset. Fine-tuning involves continuing the training process of the pre-trained model on our specific dataset, allowing the model to adapt to the genre and style of the short stories.
- **Parameters:** Use a learning rate of 5e-5, with a batch size of 16, for 4-8 epochs. Employ a linear scheduler for learning rate decay.
- **Special Tokens:** Introduce special tokens to mark the beginning and end of the story body and its ending. This helps the model recognize the structure within the input data.

4.2.2 Model 2: State-Space Model (SSM)

- **Pre-trained SSM Mamba Model:** Fine-tune Mamba (Gu and Dao, 2023) on our dataset of short stories, focusing on teaching the model to understand how narratives evolve and conclude. This step customizes Mamba's general capabilities to our specific task of generating story endings.

- **Fine-tuning Parameters:** Use a learning rate of 5e-5, with a batch size of 16, for 4-8 epochs. Employ a linear scheduler for learning rate decay

- **Special Tokens:** Incorporate special tokens or markers to delineate different sections of the stories (beginning, middle, end) and significant narrative shifts. This can aid the model in understanding the structure of narratives.

Rationale for SSM: Opting for a state-space model like Mamba leverages recent advancements in NLP to capture the temporal and dynamic aspects of story progression, which is crucial for generating coherent and contextually relevant story endings.

- **State Representation:** Define the states of our narrative. In the context of SSMs, a state could represent various elements of the story (e.g., plot points, character development stages) encoded in a vector form.

- **Dynamics Model:** This component models how the story progresses from one state to another. It would learn the transitions that typically occur in narratives from the beginning towards the ending.

- **Observation Model:** Defines how the observed outputs (e.g., sentences or paragraphs of the story) are generated from the hidden states. This model helps in generating the text of the story ending based on the final states reached by the dynamics model.

4.3 Training Procedure

- **Environment:** Train the model using a GPU-accelerated environment (GreatLakes) to handle the computational load efficiently.

- **Regularization:** Implement dropout with a rate of 0.1 to prevent overfitting.

- **Loss Function:** Use Cross-Entropy Loss to calculate the difference between the generated endings and the actual endings, optimizing the model's ability to predict the next token accurately.

- **Optimization:** Utilize the AdamW optimizer for adjusting model weights, reducing the loss over training epochs.

4.4 Post-processing (Optional)

Text Polishing: Implement a post-processing step to adjust grammar, punctuation, and style, ensuring the generated text is polished and ready for presentation or further analysis.

4.5 Model Deployment on HuggingFace

HuggingFace models has a lot of models fine-tuned for story-writing task based on the LLM GPT-2 architecture but there is none as of now which is fine-tuned on Mamba SSM. On coming up with a reasonable performing fine-tuned Mamba-based SSM model, we will also put it up for use on HuggingFace for the active NLP community and contribute to Open Source as well, as a culmination of the project and the course.

5 Evaluation and Results

Evaluating story-ending generation requires a nuanced approach to gauge both the technical and creative quality of generated narratives. Therefore, we introduce here various evaluation methods which we plan to test our model on, encompassing automated metrics like BLEU, ROUGE, METEOR, BERTScore, and perplexity, which quantitatively measure aspects such as n-gram overlap, semantic similarity, and fluency. Additionally, human evaluation will be important for looking into coherence, creativity, and emotional impact that automated metrics might overlook. Together, these evaluation strategies will provide us with a comprehensive assessment framework, ensuring a balanced analysis of a model's ability to generate compelling and contextually fitting story endings.

As an initial baseline approach, we used a random selection model. First, we created a fifth-sentence database obtained by pulling directly from each short story in the original dataset. Then, for each observation in the data, we randomly selected a concluding sentence from the fifth-sentence database and evaluated the predictions against the ground truth sentences using BERT,

METEOR, BLEU, ROGUE, and Perplexity. The average metric results for all 98,161 observations are listed in the Table 2.

As a second baseline, we used a variety of multigram models, specifically with orders 2, 4, 7, and 10. Our final evaluation was performed on the 10-gram model. The multigram models used a limited number of contextual characters to predict the following character. Using a specified contextual window, the models trained on all of the existing stories in the database and predicted 100 sentences that ideally followed the grammatical structure and subject matter of the training data. The predicted sentences were compared against 100 real sentences from the database. The results of the 10-gram model's performance are listed in Table 2.

Evaluation Metric	Baseline	
	Random Selection	N-gram (n=10)
BERT	0.869	0.852
METEOR	0.083	0.053
BLEU	0.001	0.000
ROUGE	0.070	0.053
Perplexity	140.054	518.247

Table 3: Average metric scores for the random selection and multigram baseline models.
*BERT score listed is F1 score and ROUGE score listed is ROUGE1.

As seen in Table 3, both baseline models had similar scores for each evaluation metric. On average, the random selection model appeared to marginally outperform the multigram model in each metric category.

Notably, both models produced fairly high BERT scores and fairly low METEOR, BLEU, and ROUGE scores. This indicates that the generated text itself is semantically similar to the rest of the stories, but not too similar contextually and with respect to ordering. The perplexity of the multigram model was much higher than that of the random selection model, indicating that the readability of the random selection model output was much better. This would make sense, considering the random selection model was choosing from a pool of pre-existing fifth-sentences from the original dataset, while the multigram model was generating the text character by character. Overall, the random selection model is the better performing baseline.

Outlined below are the evaluation metrics and their significance in the context of our baseline models as well as final implementation.

5.1 BLEU Score (Papineni et al., 2002)

To apply the BLEU score for evaluating a story-ending generation task where we have the actual 5th sentence of the story and a generated 5th sentence, we'll compare the generated sentence against the actual sentence using the BLEU metric. BLEU will quantify how close our generated ending is to the actual ending based on n-gram overlap.

5.2 ROUGE (Lin, 2004)

The ROUGE Score and its variants (Rouge-N/L/W) facilitate evaluation by measuring the overlap of n-grams, the longest common subsequences, and skip-bigrams between the generated text and a set of reference texts. For story ending generation, ROUGE can assess the extent to which key phrases and narrative elements in the generated story ending align with those in the actual story ending, offering a quantitative measure of narrative fidelity.

5.3 METEOR (Banerjee and Lavie, 2005)

In the context of story ending generation, METEOR provides a nuanced assessment of how well the generated ending captures the meaning and fluency of the actual story ending, taking into account paraphrasing and flexible expression.

5.4 BERTScore (Zhang et al., 2019)

BERTScore leverages the contextual embeddings from models like BERT to compute the semantic similarity between the generated text and the reference text. By comparing the cosine similarity of embeddings for matched words, BERTScore offers an evaluation of semantic congruence between the generated story ending and the actual ending, highlighting the model's ability to generate semantically relevant and contextually appropriate narrative conclusions.

5.5 Perplexity (Jelinek et al., 1977)

Perplexity measures the uncertainty of a language model in predicting the next token, providing an indication of the fluency and naturalness of the generated text. For story ending generation, a lower perplexity score on the generated ending suggests that the text is more predictable and fluent, reflecting the model's linguistic competency in crafting coherent and contextually fitting narrative closures.

5.6 Human Evaluation

Despite the advancement in automated metrics, human evaluation remains indispensable for assessing creative text generation tasks such as the story ending generation. Human judges - the authors of this work - Divya and Divyam will each evaluate 100+ one sentence generated endings for coherence, narrative satisfaction, creativity, emotional impact, and grammatical correctness, looking deeply into the qualitative aspects of story generation that automated metrics cannot fully capture.

6 Work Plan

6.1 Accomplished So Far:

1. Dataset well collected and curated.
2. Baseline Results are well tabulated and are available as a yard-stick to compare against.
3. Metrics have been implemented for baseline, so metrics in context understanding is achieved.

6.2 Intend to Accomplish Further:

- **Days till Final Report Submission on April 26 - 17 days**
- **Days 1-2:** As outlined in the methodology, We will pick up preprocessing as first step and do it to have a common preprocessed data available to both authors.
- **Days 3-7:** Both authors in parallel finetune Model 1 (Pre trained GPT-2 or GPT-3.5) and Model 2 (Mamba), train on ROCstories data and observe the results.
- **Days 8-10:** Buffer for hiccups while implementing finetuning the two models.
- **Days 11-13:** Post processing and attempt at training on longer stories and Human Judgement Metrics. Each author will evaluate 100+ one sentence generated endings for coherence, narrative satisfaction, creativity, emotional impact, and grammatical correctness.
- **Days 14-16:** Report Work and possible deployment of models (if decent results obtained) on HuggingFace Models Space. But deployment is an additional task and we won't like to commit on this as a deliverable. Report writing and completion is a promised deliverabe for this timeline.

Probable Pitfalls:

- **Experimenting with Mamba SSM for the task:** It could be possible that the Mamba model might not be able to be fine-tuned properly on the data as there could prop up unforeseen roadblocks. In that case, we would record them in the report and draw our learnings from them. However, to re-emphasize, every attempt will be made to get the Mamba model fine-tuned on the dataset and get it working as it is a learning curve for the authors as well.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the roc story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Qian Li, Ziwei Li, Jin-Mao Wei, Yanhui Gu, Adam Jatowt, and Zhenglu Yang. 2018. A multi-attention based neural network with external knowledge for story ending predicting task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1754–1762.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. 2017. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 52–55.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.