Multimodal Generative Models for Storytelling

by

Eden Bensaid

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2021

Author .....
Department of Electrical Engineering and Computer Science
01/29/2021

Certified by .....
Jacob Andreas
Assistant Professor
Thesis Supervisor

Certified by .....
Hendrik Strobelt
Research Scientist
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Multimodal Generative Models for Storytelling

by

Eden Bensaid

Submitted to the Department of Electrical Engineering and Computer Science
on 01/29/2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Storytelling is an open-ended task that entails creative thinking and requires a constant flow of ideas. Generative models have recently gained momentum thanks to their ability to identify complex data's inner structure and learn efficiently from unlabeled data [34]. Natural language generation (NLG) for storytelling is especially challenging because it requires the generated text to follow an overall theme while remaining creative and diverse to engage the reader [26].

Competitive story generation models still suffer from repetition [19], are unable to consistently condition on a theme [51] and struggle to produce a grounded, evolving storyboard [43]. Published story visualization architectures that generate images require a descriptive text to depict the scene to illustrate [30]. Therefore, it seems promising to evaluate an interactive multimodal generative platform that collaborates with writers to face the complex story-generation task. With co-creation, writers contribute their creative thinking, while generative models contribute to their constant workflow.

In this work, we introduce a system and a web-based demo, FairyTailor[1], for machine-in-the-loop visual story co-creation. Users can create a cohesive children's story by weaving generated texts and retrieved images with their input. FairyTailor adds another modality and modifies the text generation process to produce a **coherent** and **creative** sequence of text and images. To our knowledge, this is the first dynamic tool for multimodal story generation that allows interactive co-creation of both texts and images. It allows users to give feedback on co-created stories and share their results. We release the demo source code[2] for other researchers' use.

Thesis Supervisor: Jacob Andreas
Title: Assistant Professor

Thesis Supervisor: Hendrik Strobelt
Title: Research Scientist

[1] available at fairytailor.org
[2] https://github.com/EdenBD/MultiModalStory-demo

Thesis Errata Sheet

Author Eden Bensaid

Primary Dept. Department of Electrical Engineering and Computer Science

Degree Master of Engineering **Graduation date** 02/17/2021

Thesis title

Multimodal Generative Models for Storytelling

Brief description of errata sheet

I want to add the Acknowledgment page. I didn't notice that it did not compile when I uploaded the final pdf. I am sorry for the reckless mistake. Several wonderful people helped me, and I have to give them the credit they deserve.

Number of pages 1 (11 maximum, including this page)

▶ **Author**: I request that the attached errata sheet be added to my thesis. I have attached two copies prepared as prescribed by the current *Specifications for Thesis Preparation*.

Signature of author Signature redacted Date 03/27/2021

▶ **Thesis Supervisor** or **Dept. Chair**: I approve the attached errata sheet and recommend its addition to the student's thesis.

Signature Signature redacted Date 4/5/2021

Name Katrina LaCurts (chair, MEng Thesis Committee) ☐ Thesis supervisor ☒ Dept. Chair

▶ **Vice Chancellor or his/her designee**:

I approve the attached errata sheet and direct the Institute Archives to insert it into all copies of the student's thesis held by the MIT Libraries, both print and electronic.

Signature Signature redacted Date 04/07/2021

Name Ian A. Waitz

Acknowledgments

Above all, I would like to thank my two advisors, Hendrik Strobelt and Jacob Andreas, for the opportunity to learn from them and for providing feedback and advice throughout our two semesters together. The advising has helped me develop design, research, and programming skills. I had not only learned a lot but also enjoyed exploring and examining new approaches. I would also like to thank the following people from the AI Interaction team at IBM. Ben Hoover for his endless help with the demo deployment and the frontend. Mauro Martino for the thoughtful design ideas that immensely upgraded users' experience. Werner Geyer for accepting and welcoming me to the IBM 6-A program. I would not have been able to achieve anything without all those mentioned above.

Contents

List of Figures

Chapter 1

Introduction

1.1 Storytelling and Co-Creation Motivation

Automated story generation strives to generate compelling stories automatically [19].A story consists of a few sentences describing a series of events [2]. Story generation introduces compelling challenges to existing Natural Language Generation models. Compared to more constrained text generation tasks, such as machine translation and summarization, which follow existing content, story text generation has an open-ended nature. It requires diversity and creativity while adhering to a continuous narrative.

Multimodal content is prevalent in social media posts, news articles, and commercials. Among the audio, videos, and pictures modalities, images are the most common modality to accompany textual content. Adding images can enrich the content and catch readers' attention. Therefore, automatically generating a multimodal story can produce more attractive results, especially for young readers and augmenting short stories.

An interactive writing platform can support writers by suggesting new ideas and continuing previous content. It can offer exciting and entertaining directions that are nevertheless relevant to the writer's writing. Giving writers full editing power to control the final story's content keeps the users engaged. Moreover, it can alleviate writers' inertia and keep them motivated and involved in writing.

1.2 Problem Definition and Challenges

A challenging aspect of story generation is sustaining long-term memory and producing coherent text within an overall theme [26]. Another major challenge is that while adhering to a general theme and tone stories must evolve their composition and progress to new directions. Current storytelling models are limited to focus on text modality [19, 51, 25, 47, 3], without incorporating another modality such as images.

Training generative models on a single type of input, task and domain often results in a lack of generalization and robustness [38, 49]. Multimodal generative models, which are capable of relating and sharing information across multiple modalities [6], can create a representation that focuses on objects and the relations among them [49]. Existing vision-and-language models [33, 49] are not trained for storytelling generation but for other downstream tasks, such as image captioning [33], alignment prediction [33], masked multimodal learning [33], bounding box prediction [49] and visual relation prediction [49].

To encourage the model to produce more abstract representations, overseeing the textual content generation with another modality can yield promising results. There is, therefore, a compelling need for a multimodal system that incorporates both to create an engaging story.

1.3 Contributions

In this work, we propose a platform for multimodal story generation. It promises to provide measurable improvements relative to existing frameworks and demonstrate the generation of visual tales through the inclusion of:

1. A vision-and-language framework.
2. Human evaluation of the proposed platform on the story-generation objective.
3. An interactive, web-based public demo to co-create stories and demonstrate previously generated stories.

Storytelling is challenging for existing natural language generation (NLG) techniques because it requires consistency to a certain topic and creativity to engage the reader. Multimodal generation has the potential to create stories that are more effective and attractive than contemporary alternatives by adding another modality to guide the text generation process. Multimodal story generation can be applied across diverse domains such as visually grounded dialog [14], instruction following [4] and interactive, fictional stories for video-games [8].

1.4 Thesis Roadmap

The thesis follows this structure:

- **Chapter 2** reviews relevant previous work in textual story generation, story visualization, and human-in-the-loop, collaborative story generation.
- **Chapter 3** details the objectives and the methods used in this work. It describes the collected data and our framework's design, from the benchmark design to the final design we established after several iterations.
- **Chapter 4** discusses the results generated by our evaluations.
- **Chapter 5** concludes the main innovation of this work and future directions.
- **Appendix A** lists the hand-picked corpus sources.
- **Appendix B** presents the user-test format.

Chapter 2

Background and Related Work

2.1 Generative Models Background

Generative models are unsupervised (or weakly supervised), probabilistic models that model the underlying distribution of the data, often to generate new, similar samples from the model approximate data distribution [34]. Generative models have been popular in the past decade and have demonstrated promising performance in real world analysis, thanks to their capacity to efficiently learn from unlabeled data [34].

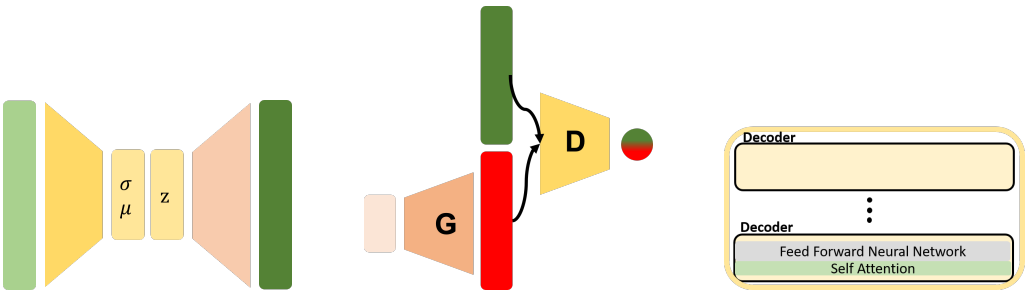
There are three popular architectures of generative models that are described in this chapter, as shown in Figure 2-1. The first two are latent variable models (LVM), statistical models that explain the structure of measurable variables with a smaller number of latent variables [18]. The latent variables are the hidden explanatory factors of the observable data points. To approximate the true distribution of the data p(x), LVMs learn the joint probability function $p(x, z) = p(x|z)p(z)$ [7]. The third architecture is an Autoregressive model, in which the current output is linearly dependent on past values and a stochastic term [22]. It is a common framework for language modeling [32], estimating the probability distribution of sequences of words to generate the next most probable word [38].

1. Variational Autoencoders (VAE): Autoencoders can compress an input to a low-dimensional latent representation z with an encoder and reconstruct the original

input from z with a decoder. Variational autoencoders can also generate new objects by learning the latent representations of the inputs as soft ellipsoidal regions (i.e., learning μ and σ vectors) rather than directly learning isolated data points (i.e., learning a vector z) [16].

2. Generative Adversarial Networks (GANs): GANs consist of two models, a generative model G that turns noise into an imitation of the training data and a discriminative model D that tries to distinguish real images from generated, fake ones. The models co-evolve during training in a framework resembling a minimax two-player game [21].

3. Transformer for Language Modeling: The original transformer model was introduced to solve the lack of parallelism in Recurrent neural networks (RNNs) computations [46]. By relying solely on attention mechanism instead of recurrence, it was able to achieve state-of-the-art performance on sequence modeling tasks while significantly decreasing previous training times [46]. The original transformer has two types of transformer blocks, an encoder block and a decoder block [46]. A modified transformer based solely on decoder blocks, was suggested to attend to longer sequences [31]. It is used in GPT-2 [38], GPT-3 [10] and Transformer-XL [13] and was able to achieve better results on language modeling objective.



(a) Variational Autoencoders (VAE)

(b) Generative Adversarial Networks (GANs)

(c) Decoder-based Transformer

Figure 2-1: Popular generative models architectures.

2.2 Related Work

There are several previous work methods for controllable story generation [19, 51, 25, 47] that aim to produce coherent text with an appealing plot. Topic conditioned models produce stories from a compact topic input [19, 51]. The advantage of the topic encoding is that it can create a compact, progressing storyline [51]. However, since the Seq2Seq model tends to focus on recently generated text and specific parts of the prompt, the plot derails from the storyline within the few (3-5) generated sentences [51]. It also frequently generates similar sentences without any sense of progression [19]. Storyline conditioned models propose tighter conditioning during the story generation by continuously directing the start to a specific ending [25, 47]. The *Unsupervised Hierarchical Story Infilling* by Ippolito et al. [25] conditions the language model on keywords that are probable to appear between the beginning and the ending of the story. The *Narrative Interpolation for Generating and Understanding Stories* by Wang et al. [47] generates several candidates and re-ranks them to take the one with the best overall coherence. Our approach mixes ideas from both topic and storyline controlled models by augmenting extracts from the dataset with automatically generated keywords and continuously re-ranking the text generation.

Story visualization architectures retrieve [39] or generate [30] images to illustrate a given story, i.e. a multi-sentence paragraph. *Coherent Neural Story Illusration* (CNSI) by Ravi et al. [39] suggests an encoder-decoder framework that can retrieve a coherent sequence of images from visualGenome [29] by predicting images' feature representations from encoded sentences and parse tree extractions [39]. StoryGAN can generate a coherent sequence of images dependent on the text by concatenating the current sentence with contextual information vector encoded from the entire story [30]. However, the text must be descriptive enough to depict the scene to illustrate in the generated images [30]. As detailed in subsection 3.4.2, we favored image retrieval since our textual content was not descriptive enough to generate valuable images. We retrieve images independently according to story pieces and use a different image dataset that corresponds better with our intended stories' genre.

Even though we did not find multimodal architectures for storytelling, combining vision and language for a joint representation is addressed by several successful models. MVAE [50] consists of one VAE model that assumes conditional independence of modalities to use product-of-experts (PoE) and reduce the number of parameters. The VAEGAN model [49] uses a VAE for text and a GAN for images on a modified multimodal objective that minimizes variational divergences [49]. As part of the pretrain-then-transfer approach, ViLBERT [33] aims to serve as a common platform for visual grounding. It has two separate streams for visual and textual inputs that interact through co-attentional transformer layers [33].

Previous approaches to generate stories suffer from repetition [19], are unable to consistently condition on a theme [51] and struggle to produce a grounded, evolving storyboard [43, 47, 25]. Story visualization often requires specific, informative text to create relevant images [30]. To address these problems in story text generation and story visualization, we offer a multimodal story generation platform that collaborates with writers. A similar interactive writing platform is STORIUM [3], an online collaborative storytelling community. However, it is intended for text completions of long stories that follow the STORIUM narrative format [3].

Our proposed multimodal story-generating framework aims to generate **creative** and **coherent** short tales by taking advantage of **multimodal** robust representations of stories, **decoder-based transformer** architecture [38, 17] and **controllable text generation**. Multimodal frameworks have been proven successful over their unimodal counterparts on various downstream tasks [49, 33]. Transformer models such as GPT-2 [38], GPT-3 [10] and TransformerXL [13] have successfully used decoder transformer blocks [31] to generate diverse, stable text. Controllable generation have encouraged generation of coherent texts. Therefore, it seems promising to compare and evaluate our multimodal generative framework on the complex story-generation task.

Chapter 3

Design & Methodology

3.1 Research Objectives

The model design process involves evaluating variations, from a baseline model described in subsection 3.4.1 to an optimized model after a few iterations, detailed in subsection 3.4.2.

The preeminent objective is to design and build a multimodal generative model that produces a **coherent and creative text and image sequence** and extends previous work on automated story generation. A coherent story follows one overall theme, and a creative story uses interesting language and is enjoyable to read.

The second objective is to **assess the model with comparable metrics** that have been commonly employed for other NLG models. Those metrics are described in subsection 4.2.1. The common issues with story generation are repetition, inconsistency and lack of progression [26, 43]. Therefore, evaluation metrics will directly assess those deficiencies.

The third objective is to create a **public demo** for users to interact with the optimized model. An accessible and easy to use web-based platform to solicit feedback on the generated stories and the schemed model.

Framed as a research question, the summary objective is: **"Can a multimodal generative platform for storytelling create coherent and creative stories?"**

3.2 Research Workflow

The general process is to collect a dataset, build a model and evaluate it. We iterated through these steps according to our findings to optimize results. The detailed steps are as follows:

1. Gather open-source and custom-collected datasets and pre-process them according to the prerequisites of our architecture.
2. Analyze datasets to examine imbalances and frequency statistics on words, sentences and image categories.
3. Fine tune a pre-trained open-source model. The model will take prompts or topics as inputs and output a sequence of text and images.
4. Evaluate the model performance with automatic metrics and the human evaluations metrics.
5. Create a public repository and demo with python and FastAPI for the back-end, and javascript and Vue.js for the front-end.

3.3 Data

3.3.1 Data collection & Processing

We tried several sources of data for each modality.

Text datasets

1. Open-source dataset - Reddit WritingPrompts [19]. This datasets has a writing prompt before each story and the stories are varied in their subjects, language and writers.
2. Manually collected dataset - Public domain children's books from Project Gutenberg. Created a fine-tuning dataset that is suitable for young readers. These books are hand-picked and cleaned before use.

To fit Reddit WritingPrompts data to our text generation style and adjust it to the Transformer model pre-requisites [38], we pre-process and clean the data as follows:

1. Trimmed stories to 1000 words.
2. Cleaned special characters and symbols from prompts and stories.
3. Removed offensive words from stories.
4. Filtered stories that were classified as having a negative sentiment.
5. Merged prompt and corresponding story to one pair and added end-of-sentence between them and at the end of the story.

From approximately 300K stories, we trained the benchmark model on 35K prompt-and-story pairs that express more positive sentiment. To predict the tonality of a given story, we used a pre-trained BERT [15] with an added GRU layer that is fine-tuned on the IMBD dataset for a sentiment analysis regression task. The model returned a value from 0-1, representing extremely negative to positive sentiment. The selected stories have a sentiment score above 0.9.

To transform Gutenberg project creative commons books for our needs, we clean the data as follows:

1. Handpicked books relevant to fairy tales generation (full list available at Appendix A).
2. Split stories to 500 tokens extracts.
3. Remove redundant new lines, offensive language, and special characters via regex patterns search and replace.
4. Removed metadata information via Gutenberg Python library.
5. Removed contents, preface, and editor notes manually.
6. To fine-tune the final model, added a generated prompt before each extract to keep a prompt-story structure and encourage controllable coherent generation.

We use approximately 9K fairy tales 500 tokens extracts to fine-tune the benchmark model for the second time, after Reddit WritingPrompts, and to fine-tune the final model for the first and only time. We do not train the final model on Reddit WritingPrompt because of the unpredictable nature of stories. Even after filtering stories, many were unsuitable for our intended young audience.

Image datasets

We tried several open-source resources to find a varied dataset that includes sceneries, people, and animals that are more closely related to fairy tales.

1. Image Retrieval: After evaluating COCO [11], Unsplash [45] and Flickr30k [52] caption-image datasets, we chose Unsplash [45] because of their relative objects' diversity and relevant landscapes nature to fairy tales.

2. Style Transfer: To achieve a coherent look of story images we fine-tuned a neural style transfer model [28] on several target images shown in Figure 3-1. The final model applies the sketch-like style to all retrieved images.

Figure 3-1: Target images used for training the Style-Transfer model [28].

3.3.2 Data Analysis

The datasets were analyzed to validate their diversity. Each text dataset was inspected to look at the number of sentences (Figure 3-3), the Part-Of-Speech (POS)

Figure 3-2: Style-transfer results on the top-left original image.

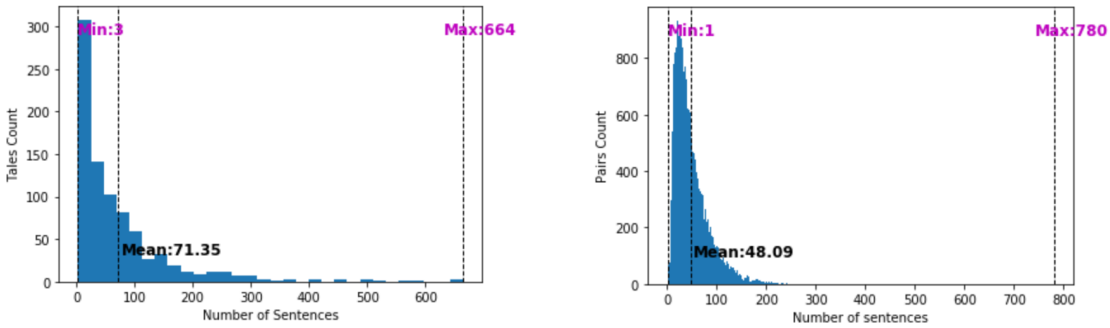
tagging (Figure 3-4) and the least frequent (Figure 3-5) and most frequent (Figure 3-6) words.

The number of sentences' distributions in Figure 3-3 verifies that our dataset mostly includes shorter stories as the ones we aim to produce. The children's stories corpus has a higher quantity of longer stories, leading to a higher mean of 71 sentences per children's story versus Reddit's mean of 48 sentences.

The Part-Of-Speech (POS) tagging distributions in Figure 3-4 displays higher concentrations of verbs and nouns than adjectives, as expected. Interestingly, in both stories' corpora, the mean number of adjectives is approximately 35% of the mean number of verbs or nouns. That high ratio might reflect the vivid, artistic nature of storytelling.

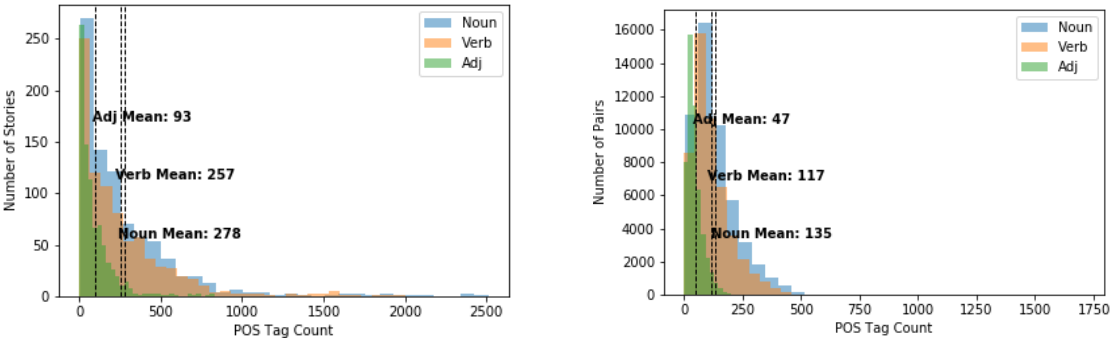
The least frequent words' distributions in Figure 3-5 shows that the Reddit corpus has a more diverse vocabulary than the children's stories corpus. Almost 50% of Reddit's vocabulary consists of infrequent words in comparison to 40% in children's stories.

The most frequent words displayed in Figure 3-6 word clouds show the characteristic old-fashioned style of children's stories. Old, upon, and shall are among the most frequent words in fairy tales, whereas terms like world, people, eyes are common in Reddit stories. Prevalent words such as I, time, and you appear in both corpora.



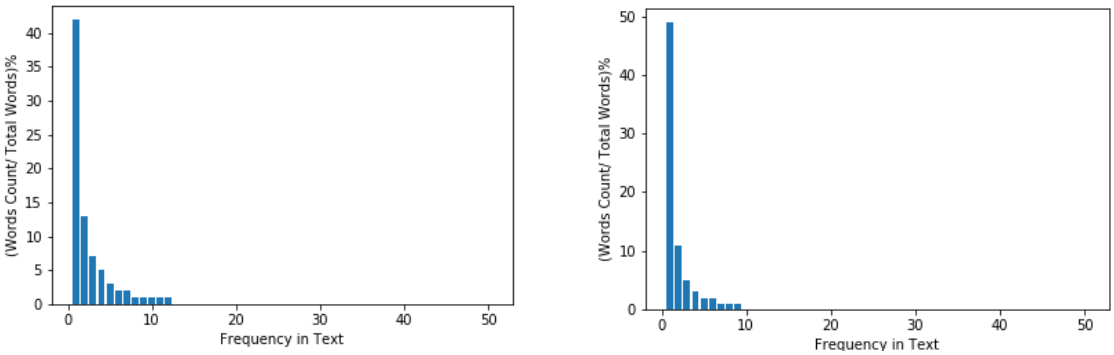
(a) Children's Stories **(b)** Reddit Prompt-Story Pair

Figure 3-3: Number of sentences in text datasets.



(a) Children's Stories **(b)** Reddit Prompt-Story Pair

Figure 3-4: Part of speech (POS) tagging of Verb, Noun and Adjective (Adj) in text datasets.

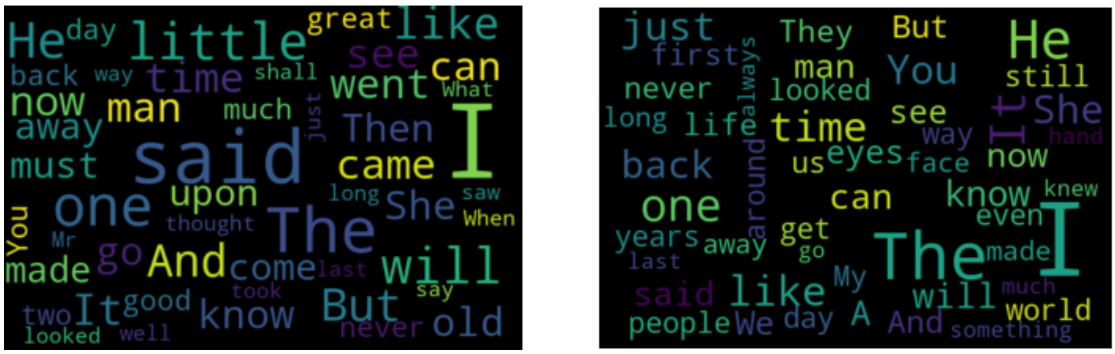


(a) Children's Stories **(b)** Reddit Prompt-Story Pair

Figure 3-5: Frequency of the least frequent words in text datasets.

3.4 Architectures

In this section, we describe two architectures that generate multimodal stories. The benchmark model suffered from repetition, inconsistency, and negative sentiments. We mitigate those flaws by changing the datasets and the framework's implementa-



(a) Children's Stories **(b)** Reddit Prompt-Story Pair

Figure 3-6: 50 Most frequents words in text datasets.

tion.

3.4.1 Benchmark Design

The benchmark multimodal generation model shown in Figure 3-7 uses the modalities sequentially, in two steps:

1. Generating coherent text:

   - Fine-tuning a pre-trained model: Two fine-tuning rounds of the GPT-2 model [38] taken from the huggingface library [48]. The first is on Reddit WritingPrompt [19] to fine-tune the model to a prompt-story template. The second is on our smaller, individually collected children's books dataset to adapt the model to a younger audience.
   - Sampling strategy: Since the decoding method can significantly affect the generated text diversity and coherency, regardless of the used model [24], we use Nucleus Sampling [24]. It was shown to encourage creativity while maintaining consistent text generation [24]. Furthermore, we experiment with *top-k random sampling* method, with $k = 50$ that was also used in the *Hierarchical Neural Story Generation* model with $k = 10$ [19], but the results were still repetitive.

2. Image retrieval from the generated text: Extract frequent nouns from the generated text and retrieve images accordingly from Flickr30K [36] caption-image

dataset. We preliminary used Flickr30K rather than Unsplash because the Unsplash dataset was not released yet. Another option we tried is to extract image categories such as animal and objects names from the generated text and use bigGAN [9] that is based on ImageNet dataset [41] to generate images. However, bigGAN limitation to 1000 categories prohibited the generation of relevant scenes.

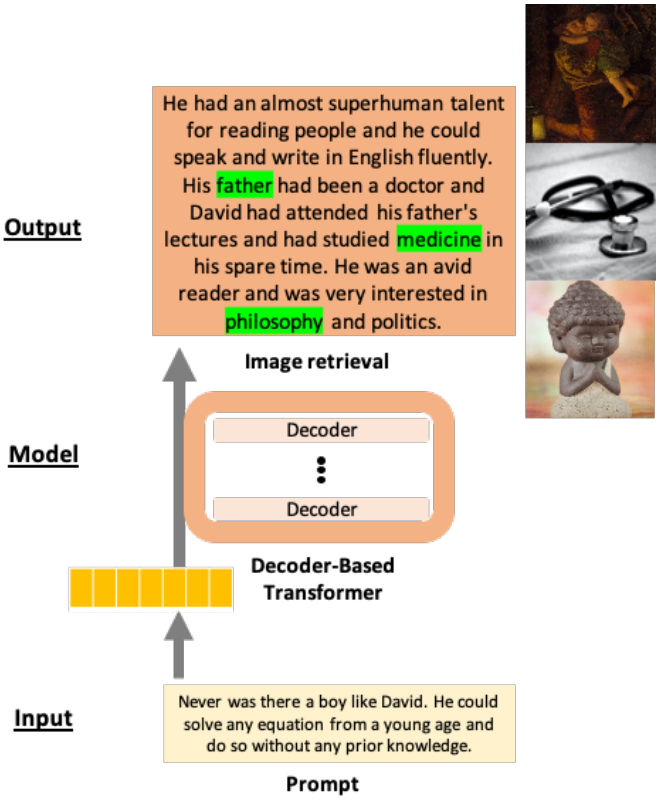


Figure 3-7: Benchmark Model Architecture: generates text from a given prompt by using a fine-tuned decoder-based transformer, and then retrieves images from Flickr dataset [36] according to key nouns.

This architecture's top new feature is the multimodal element, incorporating text and images to create a graphical story. The images guide the text generation process by re-ranking the generated samples by how coherent and relevant the retrieved images are. Another feature is the automatic text ranking performed after text generation that tests the generations' readability, diversity, and sentiment.

Testing the benchmark model revealed a few flaws that we approached in our final

Provided prompt: I was wearing my flowery jammies when suddenly a cockroach fell right on top of my head. Even though I was inside, people down the street heard me scream.

I was standing in front of the little school with the white fence, and I heard a noise in the distance. It sounded like a bell. I looked up and saw a big red rooster, about five feet long, standing right in front of me. It peered at me with an enormous mouth and a big black eye. "Well, how do you like that rooster, Miss?" the rooster asked me. I didn't know what to do. I didn't have any friends, but I did have a neighbor who was very nice. I lived in the neighborhood called Parson's Point, and we lived there on a farm that I built in the woods. The rooster looked at me with its enormous mouth and pointed at me. "That's my rooster" I said. "What do you think it is?" the rooster asked. "A rooster!" I said. "it is a rooster" said the rooster.

Figure 3-8: Benchmark model: Example of a generated story

model:

1. The text completions are often repetitive, incoherent, inappropriate, and dark.
2. The independently retrieved images are inconsistent (might get a different female figure each time)

3.4.2 Final Design

To improve upon the benchmark generation, we implemented the following:

1. Improving Text:
    - Re-Ranker Metrics: Significantly develop the ranker's role to score texts according to their readability, positiveness, diversity, simplicity, coherency, and tale-like manner. Uses min-max normalization (3.1) to rescale each

feature across all generated texts so that all features contribute equally.

$$scaled\_scores = \frac{scores - \min(scores)}{\max(scores) - \min(scores)} \tag{3.1}$$

– Readability: Calculates the length of sentences and length of words to estimate how complex the text is.

$$readability = 0.5 * letters\_per\_word + words\_per\_sentence \tag{3.2}$$

Where *letters_per_word* and *words_per_sentence* are equal to -10 if number of words and number of sentences are zero respectively. The 0.5 multiplier given a higher rank to the number of words per sentence.

– Positive Sentiment: Uses SentiWordnet [5] to compute the positivity polarity. SentiWordnet assigns sentiment scores to each WordNet [20] synonym group. WordNet is popular for information retrieval tasks and does not require pre-training. Since we do not have a supervised sentiment dataset for tales, SentiWordNet predictions were more accurate than neural nets trained on different datasets.

– Diversity: Calculates the fraction of unique words from the total number of words.

$$diversity = \frac{\text{len}(\text{set}(filtered\_words))}{\text{len}(filtered\_words)} \tag{3.3}$$

filtered_words are word tokens that exclude stop words (e.g., at, in, is) and punctuations. Score is equal to zero if there are no filtered words.

– Simplicity: Calculates the fraction of tale-like characteristic vocabulary in the given text.

$$simplicity = \text{len}(\text{set}(filtered\_words) \cap most\_freq\_words) \tag{3.4}$$

most_freq_words are precalculated to represent seven precent of the most frequent words in the collected Gutenberg fairy tales corpus.

- Coherency: Calculates the Latent semantic analysis (LSA) similarity within the story sentences compared to the first sentence. In particular, the calculation includes three steps:
  * Computing the LSA embedding of the tf-idf document-term matrix per extract.
    $transformed\_sentences = embedder(texts\_sentences)$
  * Computing the cosine max similarity per extract.
    $similarity = cosine\_similarity(transformed\_sentences)$
  * Computing the final similarity score by comparing the first sentence to the rest of the sentences.
    sum(*similarity*[0][1:])
- Tale like: Computes the KL divergence loss between preset GPT-2 and fine-tuned GPT-2 generated text's prediction scores. A higher score is better since it usually implies that the text is more similar to the fine-tuned distribution and different from the preset GPT-2 distribution. The computation consists of the following steps:
  * Tokenizing and encoding the text to *tokens_ids* to prepare it for forward pass.
  * Computing the logits of the present model *logists_preset* and of the fine-tuned model *logits_finetuned* with forward pass on *tokens_ids*.
  * Returning the difference score according to the KL-diveregence loss of the two models logits.
    torch.nn.KLDivLoss(logSoftmax(*logists_preset*),
    softmax(*logits_finetuned*)).

2. Improving Images:

- Image Generation: We tried two open-source models for text to image

synthesis, BigGAN [9] and stackGAN [53]. These models accept the generated texts as their input, but since the generated text is not descriptive of a scene, generated images are often noisy, as displayed in Figure 3-9.

Figure 3-9: Examples of StackGAN generations for the caption: "several men standing outside of small airplane with man retrieving luggage from cart."

- Image Retrieval Method and Dataset: Per-retrieval compute the Cosine Similarity of the texts noun chunks' LSA and the images captions' LSA. Return the corresponding images of the highest-scoring captions.

3. Improving Visual Story Generation to Generate a Text-and-Images Story:

- Consistency metric: Re-rank top generated stories by inner story coherency. Story consistency is calculated by summing the KL divergence difference of ResNet [23] classification predictions of image pairs. A lower score indicates a smaller difference, which is better.
- Re-Ranker Frequency: To keep relevant text generation despite a longer story generation, we re-rank after each end-of-sentence token. By re-ranking, we only keep the better half of the generation and filter out the rest.

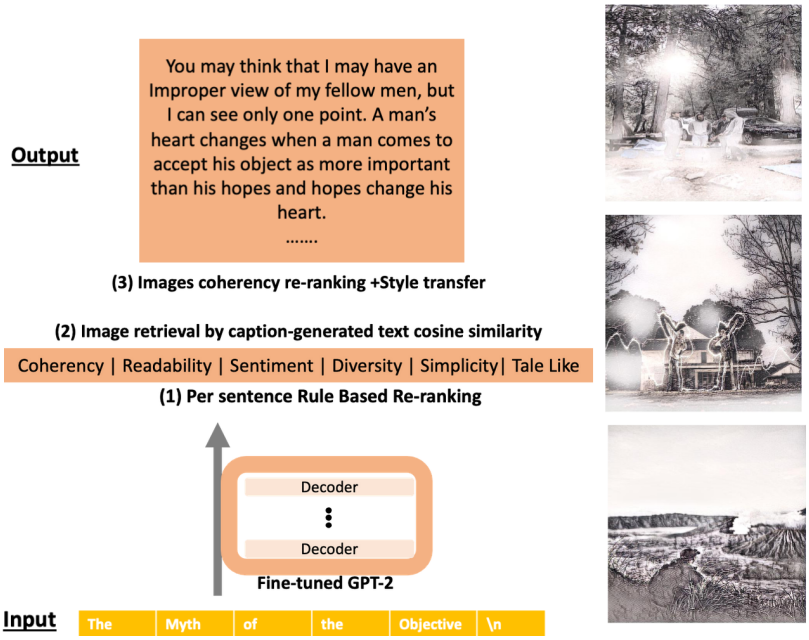


Figure 3-10: Final Model Architecture: generates text while re-ranking, retrieves images from Unsplash dataset [45], applies style transfer [28] and then re-ranks stories according to story's visual consistency.

3.5 Demo

We introduce a web-based demo, FairyTailor[1], for human-in-the-loop visual story co-creation. Users can create a cohesive children's story by weaving generated texts and retrieved images with their input. With co-creation, writers contribute their creative thinking, while generative models contribute to their constant workflow.

To our knowledge, this is the first dynamic tool for multimodal story generation that allows interactive co-creation of both texts and images. It also allows users to give feedback on co-created stories and share their results. We release the demo[2] for other researchers to quickly deploy their work and user-test any story generation model.

[1]available at fairytailor.org
[2]available at https://github.com/EdenBD/multimodal-storytelling-gan

Provided prompt: The Truth is Written in the Stars

There were once upon a time, long before the writing of the Epistles was known, a mighty kingdom whose name was in every hand of the people a kingdom of stars. And among them, in the midst of the shining city of Cyrene, was a beautiful maiden, so long ago as the time of the poet, when the world was but half its present size. Her name, for which it was named, was Aurora. But she was not the only one of the stars in the sky in whom the light of prophecy had fitted a sweet spot.
She was loved and respected by the entire race of mortals, a godlike beauty, and an angelic beauty, as is shown in the following letter. " Aurora loved you very much.“
"I dare say that she did," sighed a most melancholy-looking fellow.

Figure 3-11: Final model: Example of a generated story

3.5.1 User Interface

Users can co-create a story by starting from scratch, a given random story, or minimal content such as a story title. Users can choose between a quick or a higher quality autocomplete version, as seen in Figure 3-13.

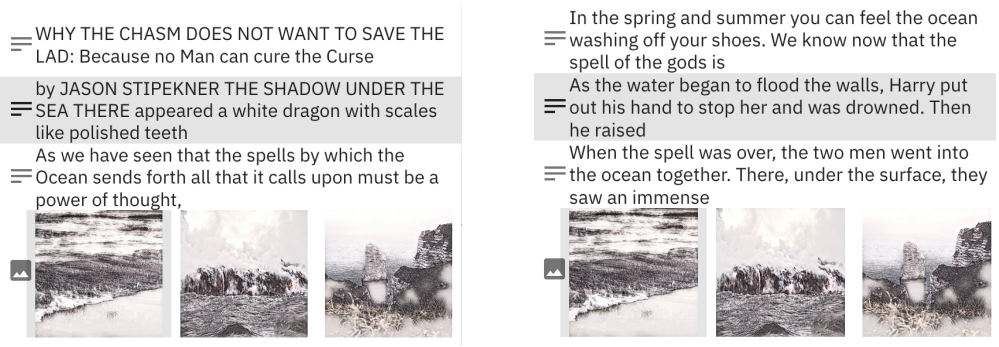
Autocomplete

The faster, more straightforward text autocomplete immediately returns the three completions generated by the fine-tuned model. It may generate empty or irrelevant completions.

High-Quality Autocomplete

Instead of generating three text completions, the framework generates ten texts, ranks them, and returns the top three. The framework scores texts according to their

readability, positiveness, diversity, simplicity, coherency, and tale-like manner. The scoring metrics are detailed in sub-section 3.4.2.



(a) Autocomplete **(b)** High-Quality Autocomplete

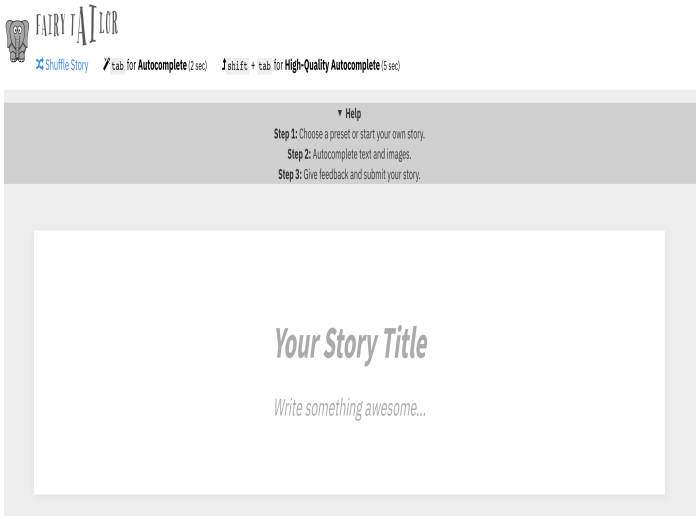
Figure 3-12: Autocompletion results to the title "The Spell under the Ocean"

Human-in-the-loop

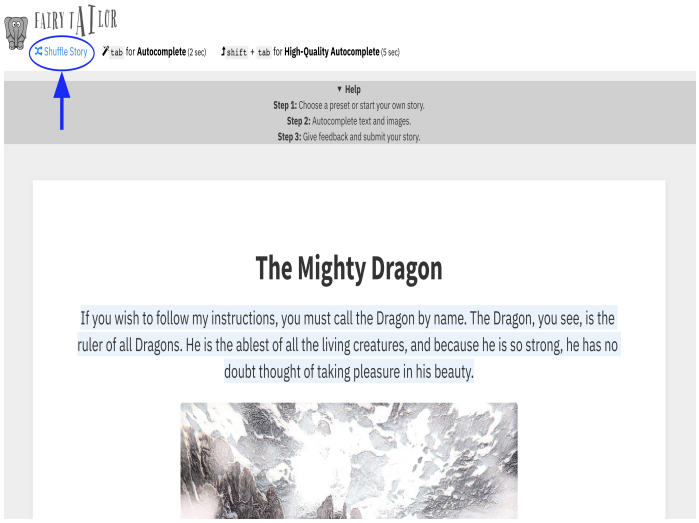
Users can add, delete, and edit the generated text and images as they wish. The generated text is marked differently than user inputted text for data collection and evaluation purposes.

User testing platform

Creative Natural Language Generation lacks reference texts and heavily relies on user evaluations instead of automatic metrics for quality checks. The demo provides a user-testing platform to share work with others and discover useful patterns quickly. Users can share their experience with a submission form, which will record their ratings, free-form feedback, and story's HTML. Researchers can use the HTML to review aspects of the generated story, such as the ratio of generated vs. user-inputted text and number of images.

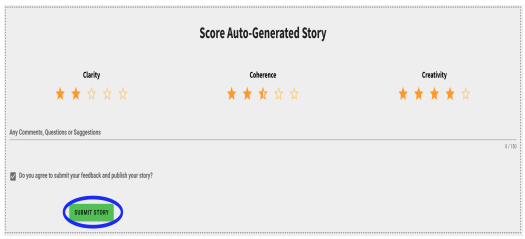


(a) Landing Page

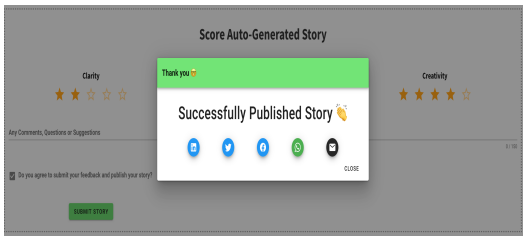


(b) Random Preset Story

Figure 3-13: Users can start writing from scratch or use preset examples



(a) Auto-Generated Story Feedback Form



(b) Published Story is Shareable

Figure 3-14: Users can publish their created stories, give feedback and share stories with others

Chapter 4

Experiment, Evaluation & Discussion

4.1 Experiment Setup

Since we are using a custom, newly collected dataset and our demo's goal is story co-creation, the most suitable evaluation practice is human evaluations [44, 3]. Automated metrics such as Perplexity [27], BLEU [35] and BLEURT [42] are unsuitable to measure creativity and coherence without reference texts. We use our FairyTailor platform and additional questions to solicit feedback on the demo interface and the generated stories. The additional questions expand on the evaluation form on the website to understand the user's journey until they hit submit. We further analyze the users' published stories to verify the efficacy of the generations. We check the ratio of generated vs. user inputted text and the text to image ratio.

4.2 Evaluation

4.2.1 Qualitative Evaluation

1. Feedback from previous work: Discussed with a few Story Generation experts that have done similar work to solicit feedback on the demo. The demo's added value from their experienced perspective is summarized below.

    - *Controllable Neural Story Plot Generation via Reinforcement Learning* [43]

co-author, Professor Mark Riedl: The main innovation is the refined, web-based interactive demo that works with writers to create stories.

- *STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation* [3] first author, Nader Akoury: The main difference is the reachability of FairyTailor to any writer, outside of the STORIUM platform, and the ability to write narratives and stories' of any structure. In addition, STORIUM autocompletes do not suggest images.
- *Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories* [12] first author, Elizabeth Clark: The significant distinctions are the writing flexibility and the intuitive options menu that enables adding images and texts.
- *Creative Help: A Story Writing Assistant* [40] first-author, Dr. Melissa Roemmele: The dominant strength is the analysis platform that can be used for evaluation thanks to the editable autocompletes and user-inputted content. The images are also novel in comparison to existing story generation platforms.

2. Human Evaluations: The structured human evaluation template is in Appendix B. It includes:

   (a) Storytelling background: Checks whether the user has written stories before, and in what context.

   (b) User feedback:

   - on the generated story, e.g., ranking the story's flow and quality.
   - on the interface, e.g., the use of autocomplete versus High-Quality autocomplete and the use of images.
   - on the overall experience, e.g., asks what the user liked and did not like.

The generated story questions are from *Predicting Generated Story Quality with*

Quantitative Measures [37]. They were designed for automated story evaluations and were previously tested on similar tasks.

We asked thirteen students and professionals (six males and seven females) to fill in our questionnaire. The interviewees include two undergraduate students in Biology Engineering and Computer Science, a Computer Science graduate student, five Computer Science Ph.D. students, two Natural Language Processing researchers, a Global Marketing Executive, a software engineer, and a technical product manager.

4.2.2 Results

Participants' insights stress what is enjoyable and what should be improved. Thirteen participants commented on the strengths and weaknesses of the interface and the multimodal framework completions after playing with the platform for a few minutes.

Demo Interface:

Overall, the participants found the demo highly engaging. A few users mentioned that the short text autocompletes that are not full sentences encouraged them to press autocomplete again, and they were less likely to delete those shorter completions.

4.2.3 Autocomplete Versus High-Quality Autocomplete:

The autocomplete and high-quality autocomplete options demonstrate the difference between the ablated version of the framework and the iterated, final one. Users who tried both indicated that the high-quality autocompletes, which take advantage of the final framework, are significantly better and were willing to wait longer for generations. It emphasized the benefits that the final model's modifications provide.

Multimodal Framework Completions:

A third of the submitted stories did not include images because users found them irrelevant to the story, or they did not think their style fits the story they had in

mind. The same happened with text completions. If the tone, vocabulary, or ideas that the completion suggested did not fit what the users had in mind, they did not incorporate it.

Published Stories:

The average scores of published co-created stories are clarity: 3.4, coherence: 3.5, creativity: 3.7. Most of the participants highlighted the beneficial creativity of the platform and its fairy tail-y nature.

User-driven changes:

We already incorporated a few of the users' ideas:

1. Since many users did not perceive the story box as editable, we updated the landing page to a blank story template instead of a preset story. As a result, the published stories we got during user-testing were very different from one user to another, which might complicate the generated stories ranking analysis.

2. Since the *"submitted form successfully"* message was not entirely clear, we added a pop-up message instead.

3. To facilitate starting from an empty sheet, we hard-coded preset titles that an autocomplete prompts.

Other suggestions that we might implement in the future:

1. Adding other modes of user interaction, such as changing the image style-transfer style.

2. Adding endings completions, since regular autocompletes do not try to summarize or direct the current story.

3. A leaderboard of the highest-scoring published stories.

4. An option for user-provided text examples to fine-tune the language model and adapt the style and probable vocabulary to the users' intended writing style.

4.3 Discussion

We found that people are excited about interactive writing and enjoyed prompting autocomplete. Some of the supportive comments include: **"I have a big interest in literature, so this is very fun"** and **"I love the highly engaging, very polished user interface"** . Flexibility was key for an enjoyable experience. People liked having control over the content, the placement of the texts and images and the timing of the completions.

4.3.1 Strengths

1. Interactive: Since our goal is co-creation of stories with human-in-the-loop, half of the questions were on the demo user interface. Users praised the ease of use and the design of the platform.

2. Creative: It is best to use the platform when users are open-minded. One of the users mentioned **"Though I did not know where my story was going initially, the autocomplete helped me find a direction"**. Prompting autocompletes is likely to generate different suggestions each time, thus helping writers guide the story.

4.3.2 Weaknesses

1. Image Retrieval Relevance: Some users did not use images because their style or content did not fit their stories. The image retrieval is restricted to the 23K images we gathered and thus do not fit every scenario.

2. Text Autocomplete Quality: Users indicated they declined to use the suggested autocompletion 50%-75% of the times they prompted it because it was repetitive or did not fit their motif. However, for users who used the High-Quality Autocomplete, the numbers were significantly lower, ranging from 0-25% of the times declining suggestions.

By analyzing people's thoughts and stories, we believe that people would use a collaborative writing platform again, especially as the quality of the text and image completions improve.

Chapter 5

Conclusions and Future Work

This chapter concludes the main innovations of this work and proposes directions for future research.

5.1 Interactive Story Co-creation

We find that participants enjoyed engaging with FairyTailor to co-create a variety of stories and would use such systems again. FairyTailor is especially beneficial for beginning writers, who find it hard to start and do not envision a specific storyboard in mind. Users mentioned that the completions' creativity helped them find a direction and maintain a continuous flux of ideas.

5.2 User Testing and Evaluation Platform

When the user submits a story, the platform saves its content along with an outlined feedback form. Researchers can quickly evaluate the ratio of generated versus inputted text and inserted images ratio with simple analytics. The platform is publicly available[1] for other researchers to deploy their work and user-test a story generation model quickly.

[1]https://github.com/EdenBD/MultiModalStory-demo

5.3 Multimodal Story Generation Framework

The image modality is novel among other story generation platforms. The images add a touch to the story and are especially prevalent in children's books. When image completions are relevant, users tend to incorporate them. Published stories that included images were ranked higher overall; users who used images praised the images' role in improving their co-generated story's quality.

5.4 Future Work

We recommend the following directions for future work in multimodal story generation:

1. User-specific completions: Currently, the autocomplete function is the same for all users. It only changes according to content. However, users' writing style and goals vary. When users have a specific storyboard in mind, the platform might never get what they envisioned and generate irrelevant completions. Incorporating an interactive feedback loop can mitigate this problem. The deletions or unused autocompletes can guide the model to the users' intentions and produce user-centered results.

2. Storyboard completions: Currently, autocompletes do not explicitly follow a storyboard and are not designated for the beginning, middle, or ending of the story. Suppose a user indicates a need to end or evolve the story by providing a goal-driven storyboard in advance or signaling while writing. In that case, it will be beneficial to have directed autucompletions that follow these cues.

3. Image generation versus image retrieval: For generality purposes, it is valuable to generate images according to input, assuming that generated images will be of high quality as retrieved images are. It will be interesting to examine Dall-E [1] from OpenAI, which demonstrated a superior ability to generate images from text descriptions.

Appendix A

Collected Gutenberg Stories' Titles

The Happy Prince, Andersens Fairy Tales, The Blue Fairy Book, The Adventures of Pinocchio, Myths Retold by Children,Household Tales, Indian Fairy Tales, Fairy Tales Second Series, MERRY STORIES AND FUNNY PICTURES, Childhoods Favorites and Fairy Stories,The Wonderful Wizard of Oz, Celtic Tales, Our Children, The Little Lame Prince, The Prince and Betty, The Adventures of Sherlock Holmes,Peter Pan,The Secret Garden, The Jungle Book, The Adventures of Tom Sawyer, A Little Princess, Little Women, Just So Stories, Moby Dick, Treasure Island, The Idiot, A Tale of Two Cities, My Man Jeeves, Sense and Sensibility, The Time Machine, Comic History of the United States, The Velveteen Rabbit, The Book of Dragons, The Snow Image, The Magical Mimics in Oz, Folk Tales from the Russian, Snow-White or The House in the Wood, Dramatic Reader for Lower Grades, A Christmas Hamper, Aesop Fables, My Fathers Dragon, The Peace Egg and Other tales, Indian Why Stories, Folk-Tales of the Khasis, The Paradise of Children, Wonder Stories, The Best American Humorous Short Stories, Hindu Tales from the Sanskrit, The Tale of Johnny Town-Mouse, The Little Red Hen, East of the Sun and West of the Moon, Among the Forest People, True Stories of Wonderful Deeds, English Fairy Tales, Simla Village Tales Or Folk Tales from the Himalayas, Japanese Fairy Tales, Plain Tales of the North, The Wind in the Willows, The Louisa Alcott Reader. A Supplementary Reader for the Fourth Year of School, A Wonder Book for Girls Boys, Tanglewood Tales, The Pig Brother and Other Fables and Stories, The Worlds Greatest Books,

Vol 3, Goody Two-Shoes, The Marvelous Exploits of Paul Bunyan, Christmas Every Day and Other Stories, The Childrens Book of Thanksgiving Stories.

Appendix B

FairyTailor User Test Template

FairyTailor, available at fairytailor.org, is a visual story co-creation platform created by MIT IBM. Users can create a cohesive story by weaving automatically generated texts and retrieved images with their input.
*Required

1. Email Address*
2. Have you written stories before? If yes, elaborate on the intended audience and the stories' structure*
3. Paste the URL of your story (created after pressing "submit story" at the bottom)*
4. Do you agree with the following statement?* (Choose one of strongly Disagree, Somewhat Disagree,Neither Agree nor Disagree, Somewhat Agree, Strongly Agree).[37]
    - Autocompletes exhibit CORRECT GRAMMAR
    - Autocompletes occur in a PLAUSIBLE ORDER
    - Autocompletes MAKE SENSE given sentences before and after them.
    - Autocompletes AVOID REPETITION
    - Autocompletes use INTERESTING LANGUAGE

- This story is of HIGH QUALITY.
- This story is ENJOYABLE.
- This story follows ONE OVERALL THEME.

5. When prompted, how often did you decline to use the suggested autocompletions?*
    - Never
    - For 25% of completions
    - For 50% of completions
    - For 75% of completions
    - Always
6. Did you use autocomplete (Tab) or High-Quality autocomplete (shift + Tab)?*
    - Mostly autocomplete (Tab)
    - Mostly HQ autocomplete (shift + Tab)
    - Both
    - Other: __
7. Please elaborate on your choice above*
8. Did you use images? Why?*
9. What did you like?*
10. What not so much?*
11. Other comments/ suggestions?

Bibliography

[1] Dall-e: Creating images from text. https://openai.com/blog/dall-e/. Published: 2021-01-05.

[2] The free dictionary: Story definition. https://www.thefreedictionary.com/story. Checked: 2021-01-28.

[3] Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484, Online, November 2020. Association for Computational Linguistics.

[4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, 2017.

[5] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *LREC*. European Language Resources Association, 2010.

[6] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy, 2017.

[7] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Feb 2017.

[8] Vincent Breault, Sebastien Ouellet, and Jim Davies. Let conan tell you a story: Procedural quest generation, 2018.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018.

[10] 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. Language models are few-shot learners, 2020.

[11] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015.

[12] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, page 329–340, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

[14] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog, 2016.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[16] Carl Doersch. Tutorial on variational autoencoders, 2016.

[17] Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anu Venkatesh, and Dilek Hakkani-Tur. Schema-guided natural language generation, 2020.

[18] B. Everett. *An Introduction to Latent Variable Models*. Springer Netherlands, 1 edition, 1984.

[19] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation, 2018.

[20] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[22] L. Harrison, K. Stephan, and K. Friston. Chapter 38 - effective connectivity. In KARL FRISTON, JOHN ASHBURNER, STEFAN KIEBEL, THOMAS NICHOLS, and WILLIAM PENNY, editors, *Statistical Parametric Mapping*, pages 508 – 521. Academic Press, London, 2007.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019.

[25] Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[26] Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. Story generation from sequence of independent short descriptions, 2017.

[27] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[29] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[30] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. *CVPR*, 2019.

[31] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences, 2018.

[32] Micha Livne, Kevin Swersky, and David J. Fleet. Sentencemim: A latent variable language model, 2020.

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[34] A. Oussidi and A. Elhassouny. Deep generative models: Survey. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–8, 2018.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[36] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, 2017.

[37] Christopher Purdy, Xinyu Wang, Larry He, and Mark Riedl. Predicting generated story quality with quantitative measures, 2018.

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[39] H. Ravi, L. Wang, C. M. Muniz, L. Sigal, D. N. Metaxas, and M. Kapadia. Show me a story: Towards coherent neural story illustration. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7613–7621, 2018.

[40] Melissa Roemmele and Andrew Gordon. Creative help: A story writing assistant. pages 81–92, 11 2015.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[42] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation, 2020.

[43] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. Controllable neural story plot generation via reinforcement learning, 2019.

[44] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. Controllable neural story plot generation via reinforcement learning, 2019.

[45] Unsplash. Unsplah dataset. https://github.com/unsplash/datasets, 2020.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[47] Su Wang, Greg Durrett, and Katrin Erk. Narrative interpolation for generating and understanding stories, 2020.

[48] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[49] Mike Wu and Noah Goodman. Multimodal generative models for compositional representation learning, 2019.

[50] Mike Wu and Noah D. Goodman. Multimodal generative models for scalable weakly-supervised learning. *CoRR*, abs/1802.05335, 2018.

[51] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling, 2018.

[52] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[53] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2017.