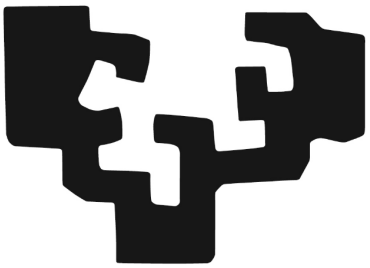
eman ta zabal zazu



Universidad del País Vasco

Euskal Herriko Unibertsitatea

FAIRYTAILOR

A model to generate storytellings

ATAI

Andoni Garrido, Giles Desmidt and Iñigo Auzmendi

INDEX:

1. Introduction:

Many models such as GPT-2 or GPT-3 have been released in recent years, which generate a text after providing a theme or a topic. The resulting text is context dependent, which makes these models appropriate for conversations or answering questions. However, for generating storytellings there are two different challenges: the text has to follow a **coherent structure** and cohesion between different sentences, and the model needs creativity. The trained model should not be based only on repeating sentences of training datasets; they have to fit with the theme of the story.

It is widely known that, in addition to text, **images** make stories more interesting for readers, enabling a visual recreation. That is exactly what FairyTailor makes; FairyTailor is a model for generating **storytellings** with both text and images. Training models for multimodal content is challenging due to the difficulty of having robust results, but some IBM researchers have created a model that works pretty well. FairyTailor can autonomously generate storytelling giving just a title, and it can also work in cooperation with writers. The model can help writers by giving them new ideas to continue the story.

Some models have been released for automatically generating short stories from a robust and compact topic. In these models it was easier to have a progression in the story, because, instead of taking into account other sentences written previously by humans, these models create every sentence within it. However in order to co-work with writers and convert in a human-in-loop model, Seq2Seq had been released. This last model focuses on the last written sentences (by writers or by the model itself) for creating new following ones. Nevertheless, the main error of this model was the lack of progression of the sentences, so that it was not focused on creating stories. Storytellers have to **progress** from the beginning till the end to create a plot and finish the story. Thus, the model has to differentiate between keywords that are used in the beginning and in the end and generate sentences using those words depending on what part of the story is in that moment. It also generates some sentences, ranking them using the overall coherence. Then, it returns the top ranks of the sentences.

Referring to image generation, this model does not generate new images, but selects the right image taking into account the theme and keywords of previous sentences. It does not generate new images because stories are usually for children and this makes sentences not descriptive enough.

2. Dataset:

In order to fine-tune the model, two major datasets have been used:

1. **Text dataset.** They have used an open source of manually written stories. The dataset used is the famous page Reddit's **WritingPrompts** thread. Herein, users post a brief description of a story called prompt and then, any writer that wants to improve skills and is interested in that prompt can write a story in full depth of around at least 100 words using that idea. For example:

- PROMPT:
  Every year, the richest person in America is declared the "Winner of Capitalism". They get a badge, and all of their wealth is donated to charity, so they have to start back up at $0
- STORY CREATED BY ‘Damptruff1’ USER:

  The CEO sat in his office. It had a deep red for a carpet, and quite a few coffee stains. The walls were painted a beautiful white, with his desk and the cabinets made out of a wood with a rich brown. He himself wore a gray suit, with a red tie and a white undershirt. He preferred a sweater and sweatpants, but today was an important meeting.

  He quickly logged onto his laptop. It was a slim device, painted in yellow and filled with the most compact electronics money could buy. He logged onto a zoom meeting, and his investors quickly joined the meeting. They were meeting about the company's stocks in relation to the Winner of Capitalism award.

  He quickly shushed their concerns, and said "It's okay. Even if I am declared the winner of capitalism, my business has a separate bank account from me. The business will stay the same, even if my bank account is emptied."

  One of the investors piped up, saying "But what if the business is declared the Winner of Capitalism?"

  The CEO replied with "It can't due to legal loopholes. Due to how the law is phrased, only people can be declared the Winner of Capitalism."

  The Investors quickly quieted down, and the CEO continued "If you continue your investment in us, we could grow our business by 50% over the next 5 years."

  The Investors were convinced eventually, and the CEO logged off his yellow laptop. He changed into his preferred outfit of sweatpants and a sweater and began his usual business.

It has also used a manually created dataset using Project Gutenberg’s free ebooks. In that database they focused on **children’s books** to train the model also for young readers.
Having 300,000 stories, they processed stories trimming them to 1,000 words, removing special characters and offensive words. Moreover, they wanted stories that transmit only positive emotions. In order to classify them by the type of emotion, they have used the BERT model, adding one fine-tuned layer using the IMDB dataset for sentiment analysis. That BERT model, once a story is given, generates a number from 0 to 1; 0 for really negative emotions, and 1 for really positive emotions. To train the FairyTailor model, stories that scored 0.9 or better in the BERT model were chosen. This was done for making FairyTailor’s stories children friendly.

After training models using two text datasets (manually picked and WritersPrompts), researchers have discovered that even filtering the WritersPrompts’s stories, the model generates stories not appropriate for young readers, so they decided to train the last model using only manually picked dataset.

2. **Image search.** In this line, popular image datasets such as COCO, Unplash and **Flickr30** have been tried. However, researchers classified objects and landscapes as the most important types of images for stories. For that reason, Flickr30, a dataset comprising a total amount of 31,000 images, was selected to use in the final model. Some examples of Flickr30 dataset:



3. System Architecture:

This technology presents two evolving architectures that are used. On the one hand, the benchmark model introduced multimodel generation but suffered from repetition, inconsistency and negative sentiments. On the other hand, after fixing those issues and limitations, the final model was presented by improving the framework and changing the data.

3.1 Benchmark Design

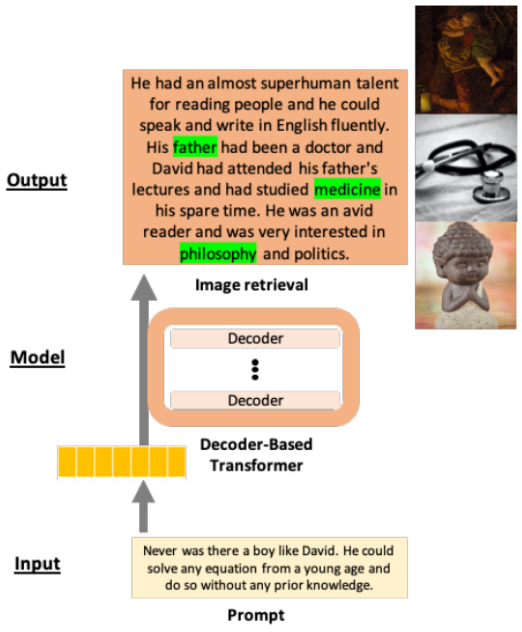
This model generates text and accordingly retrieves images. The architecture introduces a novel multimodal element. The images guide the text generation process by re-ranking the generated story samples by how coherent and relevant the retrieved images are. Another feature is that tests the generations' readability, diversity, and sentiment.

Two fine-tuning rounds were made of the GPT-2 model [Radford et al., 2018] using the huggingface library [Wolf et al., 2019]:

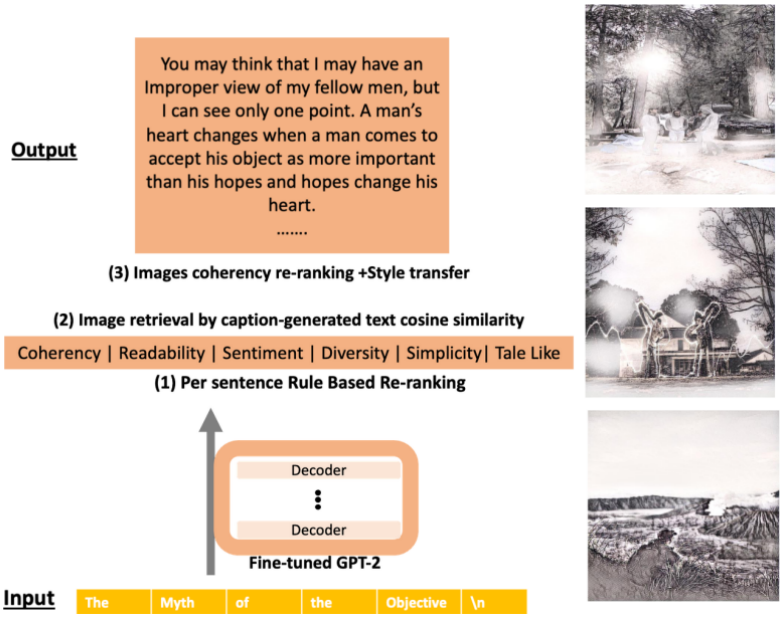
1. Reddit WritingPrompt [Fan et al., 2018] to fine-tune the model to a prompt-story template.
2. Adapt the model based on individually collected children's books dataset.

In order to cheer creativity while the text generation, they tested *top-k random sampling* method (*k = 50*) that was used in the *Hierarchical Neural Story Generation* model (with *k=10*) [Fan et al., 2018]. However, the results were repetitive and finally, Nucleus Sampling [Holtzman et al., 2019] was used.

For image retrieval, the benchmark architecture extracts frequent nouns from the generated text to retrieve corresponding images from Flickr30K [Plummer et al., 2017] image dataset.



3.2 Final Desing



Taking into account the flaws mentioned in the first version, the following improvements are proposed.

3.2.1 Text Modality

Several re-ranker metrics have been added to significantly increase the ranker's role and score texts according to their readability, positiveness, coherency and tale-like manner. The re-ranker computes the minmax normalization (1) to rescale each feature across all generated texts so that all features contribute equally.

$$scaled\_scores = \frac{scores - \min(scores)}{\max(scores) - \min(scores)} \quad (1)$$

Moreover, the re-ranker frequency has been increased. To maintain a coherent text generation, they re-rank after each end-of-sentence token. This are the features that are taken into account:

- **Readability** calculates the length of sentences and length of words to estimate how complex the text is.

  *readability = 0.5 * word_chars + sent_words*

- **Positive Sentiment** uses SentiWordnet [Baccianella et al., 2010] to compute the positivity polarity. SentiWordnet assigns sentiment scores to each WordNet [Fellbaum, 1998] 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.

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

- **Simplicity** calculates the fraction of tale-like characteristic words in the given text.

$$simplicity = \text{len}(\text{set}(filtered\_words) \cap freq\_words)$$

freq_words are precalculated to represent seven percent 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. The calculation includes three steps:
    1. Computing the LSA embedding of the tf-idf document-term matrix per *sentence. embeddings = embedder(text_sentences)*
    2. Computing the pairwise cosine similarity for each sentence against all other *sentences. similarity = cosine_similarity(embeddings)*
    3. 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 a preset GPT-2 and a fine-tuned GPT-2 generated texts' prediction scores. The computation consists of the following steps:
    1. Tokenizing and encoding the text to *tokens_ids* to prepare it for the forward pass.
    2. Computing the logits of the present model *logists_preset* and of the fine-tuned model *logits_finetuned* with forward pass on *tokens_ids*.
    3. Returning the difference score according to the KLdiveregence loss of the two models logits:
    torch.nn.KLDivLoss(logSoftmax(*logists_preset*), softmax(*logits_finetuned*)).

3.2.2 Image Modality

Three open-source implementations for text to image synthesis are evaluated: BigGAN [Brock et al., 2018], stackGAN [Zhang et al., 2017] and Dall-E [Ramesh et al.,2021]. Due to the fact that image generation times were significantly longer than image retrieval they remained with the image retrieval method. One GPU image generation time ranged from 4-30 seconds per image versus image retrieval, which took 0.5-2 seconds per image.

To compute the similarity between text and images, the cosine similarity of the text embeddings and the images embeddings are computed. The computation returns the images' ids of the highest-scoring images.

To achieve a coherent look of story images we fine-tune a neural style transfer model [Johnson et al., 2016] on several target images shown in this image:

To improve overall story generation, the framework can generate multiple stories and rank them by their images' consistency. The images' consistency metric is calculated by summing the KL divergence of ResNet He et al. [2015] classification predictions of image pairs. A lower score indicates a smaller difference, which is better.

4. **Prototype "FairyTailor"**

FairyTailor is a user interface to access the final multimodal framework and allow story co-creation. A human writer can start in multiple ways: from scratch, by using a random story primer, or by entering minimal content such as a story title. Fairytailor then offers various modes of autocomplete to assist the writer:

- **Autocomplete:** The faster, more straightforward text autocomplete immediately returns the three completions generated by the fine-tuned model.
- **High-Quality Autocomplete:** Instead of generating three text completions, the framework generates ten texts, ranks them, and returns the top three.
- **Human vs AI edits:** Writers 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.

5. **Demo**

Before Autocompletion:

The blue Knights

"I thought it was the blue eyes!"

After Autocompletion:

The blue Knights

"I thought it was the blue eyes!" said the young princess.
"You know," said the old witch, "that when a dog has his eyes cut out he becomes one of the black ones." So she called on the dogs of those times and showed them pictures of King Arthur.

6. Evaluation and Conclusion

We can conclude that this technique of storytelling is well advanced and that the technology is not standing still. Every year there is an evaluation and things are adjusted to improve programs like FairyTailor. Over time, these programs are used more and more often because they work so well and efficiently. With the help of these techniques, writers can take their stories to a new level and their creativity extends into other areas. The mixture between the use of this technique and the mind of the writer behind the story is the ideal mix to get a perfect story.

7. References

- **Eden Bensaid, Mauro Martino, Benjamin Hoover, Hendrik Strobelt [FairyTailor: A Multimodal Generative Framework for Storytelling] 2021** https://arxiv.org/abs/2108.04324

- **Eden Bensaid's github**. https://github.com/EdenBD/MultiModalStory-demo

- **FairyTailor prototype.** https://fairytailor.org/

- **Reddit's WrittingPrompts thread.** https://www.reddit.com/r/WritingPrompts/

- **Sonali Fotedar , Koen Vannisselroij , Shama Khalil , Bas Ploeg [Storytelling AI: A Generative Approach to Story Narration] 2020** http://ceur-ws.org/Vol-2794/paper4.pdf

- **Radford, J.Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2018. URL** https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
- **T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers:** State-of-the-art natural language processing. ArXiv, abs/1910.03771, 2019.
- **A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation, 2018.**
- **A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration, 2019.**
- **A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities:** Collecting region-to-phrase correspondences for richer image-to-sentence models. IJCV, 123(1):74–93, 2017.
- **Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, LREC. European Language Resources Association, 2010. ISBN 2-9517408-6-7.** URL http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf
- **Fellbaum. WordNet: An Electronic Lexical Database. Bradford Books, 1998.**
- **Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018.**
- **Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2017.**
- **Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation, 2021.**
- **Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016.**
- **He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.**