Storytelling AI: A Generative Approach to Story Narration

Sonali Fotedar[1*†], **Koen Vannisselroij**[2†], **Shama Khalil**[1†], **Bas Ploeg**[3]

[1]Eindhoven University of Technology, The Netherlands
[2]University of Amsterdam, The Netherlands
[3]Greenhouse Group B.V., The Netherlands
{s.fotedar, s.n.khalil}@student.tue.nl, koen.vannisselroij@student.uva.nl

Abstract

In this paper, we demonstrate a Storytelling AI system, which is able to generate short stories and complementary illustrated images with minimal input from the user. The system makes use of a text generation model, a text-to-image synthesis network and a neural style transfer model. The final project is deployed as a web page where a user can build their stories.

1 Introduction

Recent advancement in the field of Deep Learning has brought us closer to the long-standing goal of replicating human intelligence in machines. This has lead to increasing experimentation of neural networks as "generative", the most prominent study being Generative Adversarial Networks [Goodfellow *et al.*, 2014]. The birth of GANs lead to several variations [Radford *et al.*, 2015] and various applications in diverse domains such as, data augmentation [Ratner *et al.*, 2017], audio generation [Yang *et al.*, 2017] and medicine [Schlegl *et al.*, 2017] amongst many.

Significant breakthroughs have also been seen recently towards empowering computers to understand language just as we do. Natural Language Processing (NLP), when combined with representation learning and deep learning, saw a spurt in results showing that these techniques can achieve state-of-the-art results in many NLP tasks such as language modelling [Jozefowicz *et al.*, 2016], question-answering [Seo *et al.*, 2017], parsing [Vinyals *et al.*, 2014] and many more. 2017 saw a landmark breakthrough when the Transformer model [Vaswani *et al.*, 2017] was introduced. This sequence-to-sequence model makes use of the attention mechanism, lends itself to parallelization, and introduces techniques such as Positional Encoding that brought significant improvement over the previous sequence-to-sequence models that make use of Recurrent Neural Networks [Sutskever *et al.*, 2014], specially in terms of scalability. The Transformer model also opened up a new way of working: transferring the information from a pre-trained language model to downstream tasks, also known as Transfer Learning. OpenAI released the OpenAI Transformer [Radford, 2018], a pre-trained Transformer decoder Language Model that can be fine-tuned for downstream tasks. The model improved on several state-of-the-art for tasks such as, Textual Entailment, Reading Comprehension and Commonsense Reasoning to name a few.

Our motivation to study generative models comes after probing[1] into the content creating process within Greenhouse. Personalised content is a growing expectation that puts pressure on professionals to create and deliver novel content. We found out that the pressure of creating new and personalised content within a time crunch leads to writers' block and lack of inspiration.

More and more industry professionals are benefiting by using artificial intelligence (AI) to help them with their processes. The success of these generative models raises an important question, *can AI sufficiently help us in our creative processes?* We try to answer this question by focusing on the applications of generative models and how they can be used in content creation. We limited our scope to writing and story telling content and created the concept of Storytelling AI as a way to experiment with various generative models to create text and image content. The idea of a Storytelling AI is to generate short stories and illustrations using minimal user input.

2 System Architecture

The idea of our Storytelling AI is to generate short stories using generative models. This is achieved by accomplishing the following three sub-goals:

1. First, the user inputs a text prompt as a seed for generating a story.
2. To support the story with visuals, images are generated that are based on the text of the story.
3. Lastly, for an all-rounded experience, the generated pictures are made to look like illustrations using neural style

[1]We conducted interviews with various Greenhouse employees working in content creation.

*Contact Author
†Work done during internship at Greenhouse Group B.V.

In: A. Jorge, R. Campos, A. Jatowt, A. Aizawa (eds.): Proceedings of the first AI4Narratives Workshop, Yokohama, Japan, January 2021, published at http://ceur-ws.org

transfer.

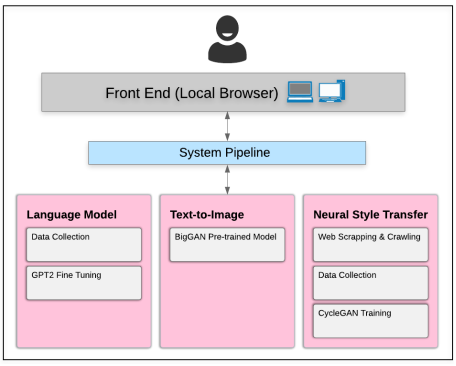

Figure 1: System Architecture Overview

Figure 1 gives a visual overview of the adopted methodology. The goals of the project are achieved by using three different generative models for the three tasks mentioned above. First, a language model is trained that learns the representation of texts in the story for the purpose of generation. Second, two text-to-image models are assessed and the best approach is adopted. Finally, a neural style transfer model is trained that learns to transfer the style of illustrated images to the images generated from the second task. The final contribution of the project is a web interface that brings these three components together where the user can build a story by generating text and images multiple times and export it in a Portable Document Format (PDF). We deploy our project by creating an interactive interface. Interested users can try the project here https://github.com/shamanoor/final-grimm-prototype.

2.1 Text Generation

The main component of our system is the generation of stories. For this purpose, we first need to model natural language by training a Language Model. Given a vocabulary of words, a language model learns the likelihood of the occurrence of a word based on the previous sequence of words used in the text. Long sequences of text can then be generated by starting with an input seed and iteratively choosing the next likely word.

Our system uses OpenAI's GPT-2 for the purpose of language modelling [Radford *et al.*, 2019]. GPT-2 is a large Transformer based [Vaswani *et al.*, 2017] language model with 1.5 billion parameters, trained on a data set of 8 million web pages called WebText. GPT-2 is built using the transformer decoder blocks with two modifications: first, the self-attention layer in the decoder masks the future tokens, blocking information from tokens that are to the right of the current token and, second, adopting an arrangement of the Transformer decoder block proposed by [Liu *et al.*, 2018]. Different sized GPT-2 models have been introduced by OpenAI. Due to the low compute capability of the available hardware, the small GPT-2 model is used in our system. To achieve our goal of generation of stories, we fine-tuned the pre-trained GPT2 language model on a data set of short stories. To construct our data set, we collected 100 short stories written by the Brothers Grimm from Project Gutenberg[2].

Figure 2: Samples generated using StackGAN and BigGAN: a) images generated by StackGAN conditioned on text description and b) images generated by BigGAN conditioned on an object class.

2.2 Text-to-Image Synthesis

The next step in our work is to generate images that complement the generated story. Text-to-Image synthesis technique was adopted for this goal. We explore two ways to do this: first, to make this process as automated as possible, we adopt the StackGAN architecture to generate the images given a text description, second, we adopt a less automated technique the BigGAN API is used to generate images conditioned on a class.

StackGAN. StackGAN was proposed by [Zhang *et al.*, 2016] to generate photo-realistic images conditioned on text descriptions. The idea of StackGAN is to decompose a hard problem into more manageable sub-problems through a sketch-refinement process, therefore using two models *stacked* on one another. The original paper used several datasets to evaluate their work, but for our system, we only use the model pre-trained[3] on MS COCO dataset [Lin *et al.*, 2014] since it is a more generalized dataset containing 80 common object categories and relates more to our problem.

BigGAN. The next technique adopted for text-to-image synthesis is less automated to aim at more controlled and realistic generations. For this purpose, the pre-trained BigGAN model from HuggingFace[4] is used. BigGAN, proposed by [Brock *et al.*, 2018], is a class-conditional image synthesis technique attempting a large scale GAN training for high fidelity natural image synthesis. The model is trained on the ImageNet dataset [Deng *et al.*, 2009] and can generate high fidelity images from 1000 classes. We use the pre-trained BigGAN deep 256, a 55.9M parameters model generating 256x256 pixels images conditioned on a class[5].

Figure 2 shows text-to-image generation using StackGAN and BigGAN. It can be seen clearly that the generations using StackGAN are vague and imprecise. Some images are

[2]https://www.gutenberg.org/
[3]https://github.com/hanzhanggit/StackGAN-Pytorch
[4]https://huggingface.co/
[5]https://github.com/huggingface/pytorch-pretrained-BigGAN

able to generate the setting of the description, for example, fields or beaches, but the overall quality of the generation is very poor. The images generated by BigGAN conditioned on a class are of far superior quality than the ones generated using StackGAN. Therefore based on qualitative analysis, we see a clear trade-off between automation and fidelity in the process of text-to-image synthesis. Since the aim is to have image generations of higher quality, we compromise on automation and use the BigGAN model to obtain better quality class-conditioned image generations. Images generated by BigGAN do not depict a whole description with multiple objects, but we settle for a comparatively higher quality generation of a single object.

2.3 Neural Style Transfer

We also aimed at generating images that look like an illustration, therefore aiming at a more all-rounded storybook feel. Therefore, the last step in our system is to have illustrated images by using Neural Style Transfer to transfer the illustration style to our generated images. We use the CycleGAN model to perform neural style transfer on our generated images. The model was proposed by [Zhu *et al.*, 2017] as an approach for learning to translate an image from a source domain to a target domain in the absence of paired examples. If X denotes images from the source domain and Y denotes images from the target domain, then the goal is to learn a mapping from X to Y.

To build our dataset, we randomly sample 6500 images from the MS COCO data set [Lin *et al.*, 2014] for training and 1000 images for testing. We further collect illustrated images by crawling through Pinterest boards relating to illustration art and fairy tale and story illustrations. We scraped 6,308 images from these web pages using `BeautifulSoup4`[6] and `Selenium`[7]. The images were manually analysed and noisy images such as non illustrated images were removed. Black and while images and images with texts were also removed. They were also randomly cropped into a 1:1 dimension ratio. We then train the CycleGAN model from scratch on these data set. Figure 3 illustrates some examples of style transfer using the trained model.

2.4 Front-end

Now that we have the main building blocks for our storytelling system, the final step is to create a pipeline of these models using a user interface. In Figure 4, we share a snippet of the user interface.

The interface allows the user to input a text prompt that the trained language model uses as a seed to generate chunks of stories. Since imperfections in the generated text are inevitable, the text can be edited to the liking of the user in the text box. Simultaneously, the user can also generate illustrated images by choosing an object class from a drop-down menu that they think would best compliment the text generated. This process requires two background steps: first, the selected object class is used as input to generate a class-conditioned image using the pre-trained BigGAN model, and

[6]https://www.crummy.com/software/BeautifulSoup/bs4/doc/#

[7]https://www.selenium.dev/

Figure 3: Neural Style Transfer: Illustration style transferred using CycleGAN on images generated by BigGAN.

second, the generated image is fed to the trained CycleGAN model to generate the image with an illustration style. These generations can be performed multiple times and added to the final story, where the user can add a story title. When the user is satisfied with the story, they can export it to a PDF file.

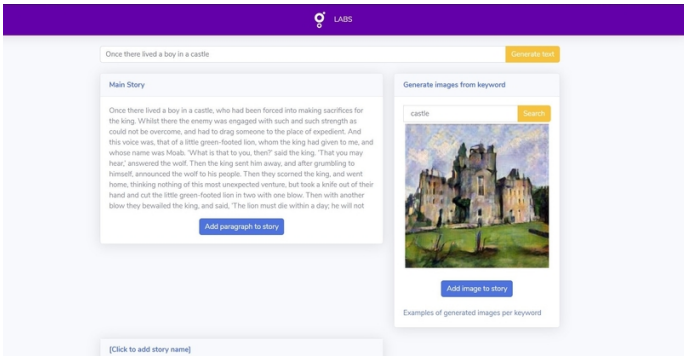

Figure 4: User Interface

3 Conclusion

In this work, we demonstrate a Storytelling AI that uses generative models to create stories with complementing illustrations with minimal user input. Our aim with this project was to study generative models and their competency in generating original content. We believe that given the advanced state of technology AI techniques can generate human-like content but it requires human intervention and supervision to a great extent. With research being conducted towards more controllable generations, we believe with a well curated data set, generative models can help conceptors in creating novel and personalised advertisement sketches, design and images.

Acknowledgments

We would like to extend our greatest thanks to Dr. Decebal Mocanu for his constant supervision and invaluable guidance throughout the course of the internship. Moreover, we would like to thank Mr. Ruben Mak and Mr. Thom Hopmans for their coaching and counselling during our internship at Greenhouse Group.

References

[Brock *et al.*, 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[Deng *et al.*, 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

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

[Jozefowicz *et al.*, 2016] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling, 2016.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[Liu *et al.*, 2018] Peter Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. 01 2018.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[Radford *et al.*, 2019] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[Radford, 2018] Alec Radford. Improving language understanding by generative pre-training. 2018.

[Ratner *et al.*, 2017] Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation, 2017.

[Schlegl *et al.*, 2017] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017.

[Seo *et al.*, 2017] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603, 2017.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[Vinyals *et al.*, 2014] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language, 2014.

[Yang *et al.*, 2017] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation, 2017.

[Zhang *et al.*, 2016] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.