Just Say No to the Boring Books

Abstract

In this project, we propose a system to generate illustrations for paragraphs from the books in a specific style based on their genres to make boring novels in our literature, interesting. The pipeline of the system consists of three main parts: a knowledge distillation-based information retrieval model to summarize a given paragraph, a clustering model to figure out the genre of the input, and fine-tuned stable diffusion model to generate an image from a paragraph summary given special style, which depends on the genre.

1 Introduction

The visuals in our childhood books are some of the most memorable components, so why do we not see them more often in adult novels as well? One may explain it as lack of style or devotion to the force of the word, or simply that advanced readers do not require illustrations. However, it is difficult to argue that an adult text with some artwork supporting the story complements a book. There are many reasons why we should use more images in books. Images can provide additional information that supplements the text. They are helpful for communicating ideas and information, and by supporting the text's material, they can enhance understanding. They can break up large blocks of text, making the book more visually appealing. They can help set the tone and atmosphere of the book so the reader doesn't lose attention. All these reasons compelled us to make a solution to make visualization of text paragraphs in novels to make them even more interesting. The architecture of our proposed pipeline can be seen in figure 1.

2 Contribution

Stable diffusion models can be useful for book writers in a variety of ways. For example, a book writer might use a stable diffusion model to predict how quickly a particular idea or concept will spread through a given population. This could be useful for determining how long it might take for a new book to become popular, or for assessing the potential reach of a marketing campaign.

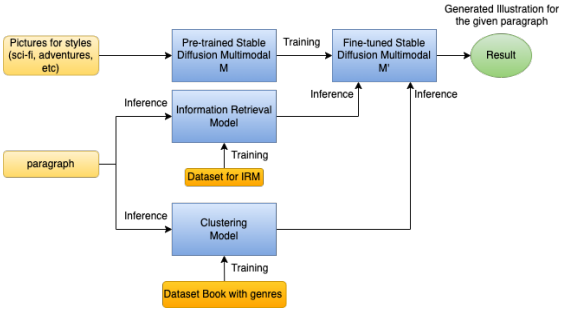

Figure 1: Architecture

Additionally, stable diffusion models can be used to analyze the spread of certain themes or ideas within a book or across multiple books within one public publish of the picture that represents a book. This could be useful for identifying trends or patterns in reader behavior, or for understanding the ways in which different ideas or concepts are received by readers.

Overall, stable diffusion models can provide valuable insights into the way ideas spread, and can help book writers to better understand their audience and the potential impact of their work. The only limitation is that the author needs to manually engineer the prompt. Our solution will automate this in a reasonable time. (Student-Teacher aka Knowledge Distillation decreases the time of producing the summary)

3 Related Work

For the clustering part, the most related work is Coll Ardanuy and Sporleder (2014). They tried to construct clusters to classify genres of books. They

used automatically extracted static and dynamic networks to perform large-scale analyses of novels, by representing them as vectors of features that can then be used to compare the novels in terms of genre and authorship. However, on our clustering part, we deal with paragraphs, not whole novels.

Text summarization (Allahyari et al., 2017) is a natural language processing task that involves generating a concise and fluent summary of a given text document. This is a challenging problem, as it requires the model to understand the content and structure of the document, as well as identify the most important information and produce a summary that is coherent and accurately reflects the original text.

There are two main types of text summarization: extractive summarization and abstractive summarization. Extractive summarization (Zhong et al., 2020) involves selecting and combining important sentences or phrases from the original document to form the summary. This is typically done using algorithms that identify important sentences based on factors such as word frequency or position in the document.

Abstractive summarization (Gupta and Gupta, 2019), on the other hand, involves generating new sentences that are not present in the original document. This is a more challenging task, as it requires the model to understand the meaning and intent of the original text and generate new sentences that accurately reflect this meaning.

There has been a significant amount of research on text summarization, with many different approaches proposed. Some common techniques include the use of extractive methods such as sentence ranking or clustering, as well as abstractive methods such as neural network-based sequence-to-sequence models (Rosca and Breuel, 2016).

Recent advances in natural language processing, particularly in the field of deep learning, have led to significant improvements in the performance of text summarization models. These models are now able to produce high-quality summaries that are competitive with human-generated summaries (Kiyoumarsi, 2015).

4 Implementation

Our system (Figure 1) consists of three main parts: Clustering, Information Retrieval and Stable Diffusion image generator.

The preprocessed paragraph will go through the Clustering model to figure out its genre which will specify the style for illustration. The same paragraph will go through the Information Retrieval model to summarize the given paragraph because some paragraphs are big, and their size will badly affect on image generator model. Finally, outputs of the previous two models, summarization and genre, will be input for Fine-tuned Stable Diffusion Multimodal, which was fine-tuned on multiple images with different styles, and which will provide the final result: generated illustration for the given paragraph.

5 Text Clustering

The clustering model is used to figure out to which genre a paragraph belongs. These genres will be used as input for the image generator model to specify which style to use. However, there is a problem with data: there are a lot of datasets with books and their genres, but there is no data with paragraphs and their genres. We can not train the model only on the whole text of books, or assign the genre of the book to a paragraph because they can differ (for example horror books can have romantic paragraphs).

5.1 Proposed Method

We decided to create the dataset by dividing downloaded books into paragraphs and cluster all paragraphs from all books using the KMeans algorithm. We assume that paragraphs with the same genre will be in the same cluster. And then assign dominant genres of books, from where these paragraphs were obtained, to these clusters.

We used 11 books for constructing clusters (2 per genre, and one additional horror book, because they are usually small) and 3 books for testing. We tried to download books that are related to a single genre for the train set.

5.2 Experiments

For transforming text to vector space we used the TF-IDF vectorizer. Before this, we removed all stop words and punctuation and changed misspells like ”don’t” to ”do not”. After the first experiments, we decided to remove all names of characters and organizations, because with character names we faced problems when one cluster consists of only one book.

Also, we tried to construct strictly 5 clusters, one cluster per genre, but we still faced problems with

the adventure book. One adventure book was in one cluster, and when we watch on top 10 words in this cluster it was clear it was because of the place where actions were. Vocabulary that is used to describe sea adventures totally differs from what is used for adventure on land, and because of the TD-IDF method vocabulary had a big effect on clustering. As the result, we had two clusters with the adventure genre and one genre without the cluster.

We decided to increase the number of clusters. We used 13 clusters for the KMeans algorithm because of the output of the Silhouette method (Rousseeuw, 1987) as shown in figure 2.

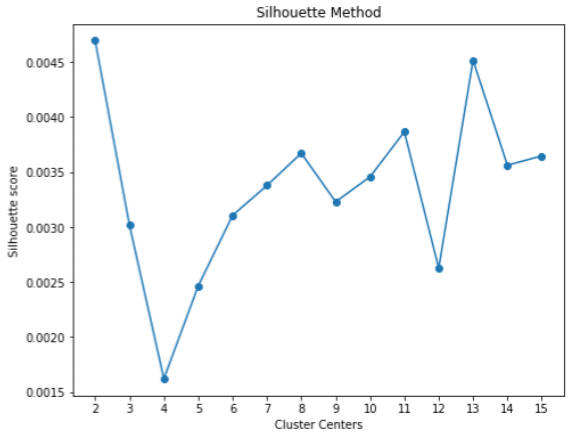

Figure 2: Architecture

5.3 Results and Discussion

We tried to cluster some new paragraphs from books that the model did not see. The model was able to figure out adventure paragraphs from a "15-year captain" that took place on the sea, however model faced problems with other adventure paragraphs that took place on the land. Instead of it, they were clustered as detective and sci-fi. We faced the same problem for another book from the test set "Three Musketeers", the model faced problems with adventure paragraphs but performed well on the romance part. As the result, our model is unable to categorize paragraphs having context specifically about adventure on land.

6 Information Retrieval

In order to generate illustrations we just need the important information from a certain paragraph because some paragraphs are big, and their size will badly affect on image generator model.

6.1 Proposed Method

As we only had paragraphs from the books but not the labels so first we needed to extract summaries out of all those paragraphs, and later train our summarizer model on that parallel data to retrieve only useful information out of the whole paragraph so that we can pass it to the stable diffusion model to get the illustrations. We used a knowledge distillation approach to first make the summaries of all the paragraphs and then auto-trained student summarization models with varying maximum summary lengths as shown in Figure 3

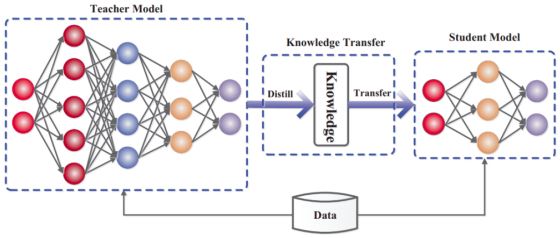

Figure 3: Architecture

6.2 Experiments

To first get the summary of each paragraph we needed to use a teacher model. We started with GPT3 (Floridi and Chiriatti, 2020). Turned out this was not a good idea as it was only outputting the extractive summarization (Zhong et al., 2020) of the input paragraph which we could achieve by simply using any statistical approach. After an extensive literature review as discussed earlier we decided to go on with BART models (Lewis et al., 2019)

BART, or Bidirectional Encoder Representations from Transformers, is a state-of-the-art natural language processing model developed by researchers at Facebook AI. It has been widely used for a variety of natural language processing tasks, including text summarization.

One of the key advantages of BART is its ability to perform well on a wide range of tasks without the need for task-specific fine-tuning. This makes it an attractive choice for text summarization, as it can be quickly and easily applied to a given summarization task without the need for extensive training.

Studies have shown that BART can produce high-quality summaries that are competitive with other state-of-the-art models. In one study, BART was compared to other summarization models on the CNN/Daily Mail summarization dataset and was

BART	Validation Metrics		
	Loss	Rouge1	Rouge2
Model 1	0.353	**52.79**	**46.13**
Model 2	0.286	42.38	35.02
Model 3	**0.227**	33.63	27.80

XSUM	Validation Metrics		
	Loss	Rouge1	Rouge2
Model 1	1.16	43.42	23.55
Model 2	**1.20**	**44.17**	**24.65**
Model 3	1.32	32.43	18.01

Table 1: Validation Metrics of Student Models

found to produce summaries that were on par with or better than those produced by other models.

We later trained multiple student models on parallel text. And based on their Rouge2 score (Schluter, 2017) we decided which one to use at the last step for the stable diffusion. Link to text summaries by BART[1]. Link to summaries by XSUM[2]

6.3 Results and Discussion

For the teacher models, we used **distilbart-xsum (R2: 21.37)**, **bart-large-cnn (R2: 21.06)** and **GPT3 text-davinci-002**, with **255M**, **406M** and **175B** parameters respectively to make labels (summaries) for paragraphs. Our try with GPT3 failed as it created very poor summaries, for the rest of the two teacher models, we trained 3 different student models varying in maximum summary lengths **(30, 50, 100)**. Each student model is optimized one among additional 5 models, to get a total 30 combinations. The validation metrics of these top 3 (Rouge2) student model per teacher can be seen in Table 1

There was no proper dataset available for book paragraph summarization task. We spent most of the time in creating and cleaning our own datasets. On top of using BART and XSUM for teacher models, we plan to use T5 Model (Xue et al., 2020) to make summaries.

7 Stable Diffusion

Fine tuning of stable diffusion is a process in which the parameters of a diffusion model are adjusted in order to improve its accuracy or performance. This is often done in order to better match the model to real-world data, or to optimize it for a specific application.

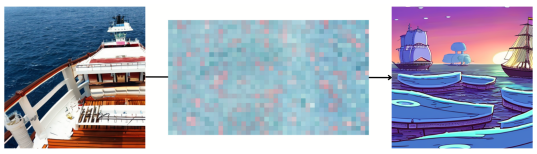

Figure 4: An illustration of contrast between the default and tuned variants of the Stable Diffusion. Adventure Style

7.1 Proposed Method

In order to create personalized styles for different book genres, we collected images from Google. This allowed us to create unique styles that correspond to each book genre. To ensure that the fine-tuning process is successful, it is important to consider certain conditions during the data collection phase. For example, we need to ensure that the colour scheme and art style are consistent across the images and that the images are not too complex or cluttered. Also, it needs carefully select a small number of images (3-5) for each style, in order to avoid overfitting and ensure that the model converges. We selected images that could be easily distinguished from each other and deliberately chose images that were representative of each genre. For example, we selected images of planets, galaxies, and blue and dark blue colours for the sci-fi genre, and images with pink backgrounds for the romantic genre. By using this approach, we were able to train the model to learn from the images and create personalized styles for each book genre. Link to Dataset [3]

With the collected dataset, we plan to fine tune the stable diffusion model in order to improve its performance on book genres. This will involve adjusting the model's parameters using techniques such as optimization algorithms or machine learning methods. Once the model has been fine tuned, we will test it on a set of known data to evaluate its performance. If the model performs well on this test data, it will be ready for use in real-world applications.

7.2 Experiments

In this paper, we present two methods for fine-tuning a pre-trained stable diffusion model: textual inversion [4] and Dreambooth (Ruiz et al., 2022).

[1] BART Dataset
[2] XSUM Dataset
[3] Image Dataset
[4] https://github.com/invoke-ai/InvokeAI

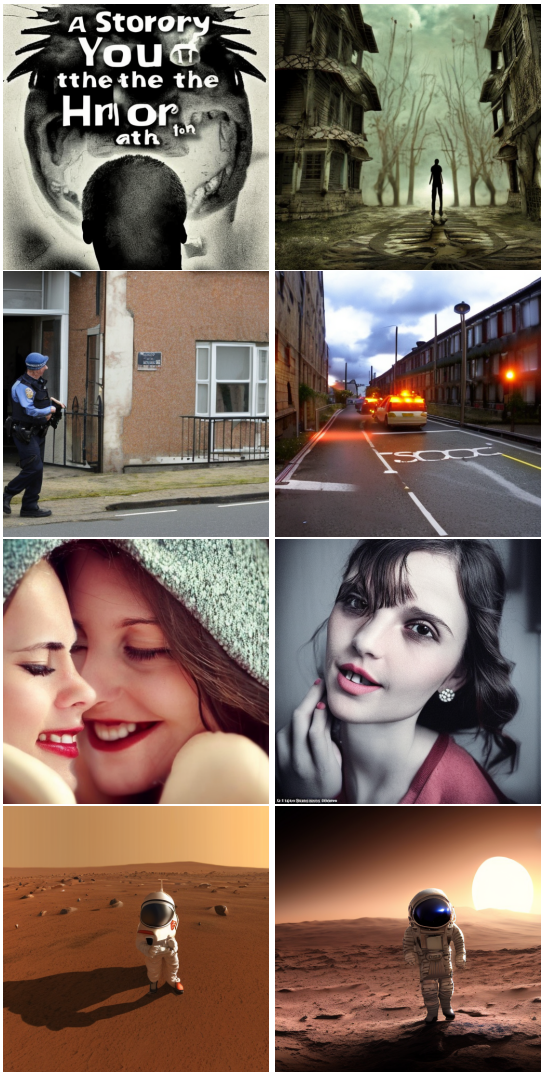

Figure 5: Default-Tuned images. Horror, Detective, Romantic, SciFi styles

The textual inversion method does not modify the initial stable diffusion model but instead creates separate embeddings for the new, fine-tuned model. This is advantageous because it prevents the original model from being modified, and allows for version control of the fine-tuned model. In contrast, the Dreambooth method directly modifies the original model, which can lead to changes in the model's behaviour. We found that the textual inversion approach was more effective and chose to use it in our experiments.

7.3 Computer Hardware

In our experiments, we found that the default configuration of the stable diffusion model required at least 20GB of VRAM for training. To make the training process more efficient, we reduced the default batch size and increased the number of workers. However, these changes did not significantly improve the training speed. We also attempted to use hardware acceleration techniques such as DeepSpeed [5] and xformers [6], but were unable to install them on MBZUAI's lab PCs. Despite following the steps in the tutorial and documentation, it took us several days to determine the optimal configuration for our project. The training time for each model varied depending on factors such as the style, number of images, and complexity of the patterns in the images, but typically took between 1.5 and 2 hours to complete.

7.4 Results and Discussion

In our experiments, we observed that using a summary of 30 words resulted in more accurate images, but the quality of the text information was reduced. In contrast, using a summary of 100 words produced more detailed text information, but the images generated by the stable diffusion model were more abstract and less obvious. These observations suggest that there is a trade-off between the accuracy of the images and the quality of the text information and that the optimal summary length will depend on the specific application and the goals of the analysis. For this experiment the optimal paramater for text summaries was 50 words. An example of image generation by default and tuned versions of stable diffusion is shown in figure 5. We applied the same text summary to both models, but varied the style invoke.

8 Limitations and Future Work

For clustering, we faced some problems with certain paragraphs in the adventure genre. Our model is unable to categorize paragraphs having context specifically about adventure on land. In future, we plan to solve this problem by using contextualized embeddings instead of just vectorized ones that we used now.

There was no proper dataset available for the book paragraph summarization task. We spent most of the time in creating and cleaning our own datasets. On top of using BART and XSUM for teacher models, we plan to use T5 Model (Xue et al., 2020) to make summaries.

Fine-tuning a stable diffusion model is memory and time expensive task. We used only one sample

[5] https://github.com/microsoft/DeepSpeed

[6] https://github.com/facebookresearch/xformers

in a batch, however with more GPU resources, we could use more samples in a batch to get better illustrations. In future, we plan to use the newly released stable diffusion 2.0 for our task.

9 Conclusion

Even it was really difficult for the KMeans to cluster paragraphs to a specific genera we still got pretty good results. The student models trained on BART summarizer performed much better than the ones from XSUM as can be see in Table 1. Certain discoveries are worth to be mentioned. A 30 word summary delivers more accurate visuals, but the text material loses quality. Although the 100-word description contains the greatest information, the stable diffusion model produces more abstract and non-obvious pictures. So 50 words is the appropriate number of words.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Mariona Coll Ardanuy and Caroline Sporleder. 2014. Structure-based clustering of novels. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 31–39, Gothenburg, Sweden. Association for Computational Linguistics.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694.

Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Farshad Kiyoumarsi. 2015. Evaluation of automatic text summarizations based on human summaries. *Procedia-Social and Behavioral Sciences*, 192:83–91.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Mihaela Rosca and Thomas Breuel. 2016. Sequence-to-sequence neural network models for transliteration. *arXiv preprint arXiv:1610.09565*.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2022. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.

A Text that was used to produce images.

Text that was used to produce images in figure 4.

- The story of a ship that was on the deck at the end of the year. (adventure)

Text that was used to produce images in figure 5.

- The story of a man who has been working with a young man in his head. (horror)
- Police considered that it might be caused by Arthur's closing his bedroom door. (detective)
- She knew that feeling and knew its signs, and saw them in Anna; saw the quivering, flashing light in her eyes, and the smile of happiness and excitement unconsciously playing on her lips. (romance)
- I wasn't expecting to be first at anything. I was the 5th crewman out of the MDV when we landed, making me the 17th person to set foot on Mars. (sci-fi)