From script to storyboard:
Using extractive summarization to generate prompts for storyboard generation

Shradha Dinesh

Abstract

This project tests the outputs of extractive summarization models on screenplays, a narrative form that natural language processing methodologies have yet to meaningfully explore. The outputs of the scene summaries generated by the project were run through the Stable Diffusion text2img model to generate an accompanying screenplay. I built an application using Streamlit to generate summaries and run them through Stable Diffusion. By human review, I found that current transformer models for extractive summarization struggle with the task of preserving essential named entity relationships in screenplays, and with shorter scenes, tend to abstract new plotlines and details that did not exist in the original text.

Introduction

The original intent of this project was to develop a product that could distill movie scripts into their scenes so that they could be illustrated by AI illustration tools. Illustration is a crucial process in film and television pre-production, and it encompasses writing, blocking, FX, sound, and cinematographic design. Creatives at each point in a production need to understand how all of the narrative and physical elements of the production come together in a scene. The goal of this project was to develop a workflow that could assist storyboard artists with preliminary design, writers with scripting, and directors with production.

In its current state, AI illustration has been discussed as a novelty product, and its use for common artistic tasks has not yet become a widely adopted practice. Discussions around AI art are concentrated in the realm of disinformation research and intellectual property. It is important to note that creatives in film/TV production are actively concerned about the role of AI technology. In May 2023, the Writers Guild of America went on strike; one of the Guild's concerns was the potential of studios to replace writing jobs with AI. The threat to illustrators is adjacent.

As the results of this project show, the production of this technology still has a long way to go before its use can be considered for use in artistic projects.

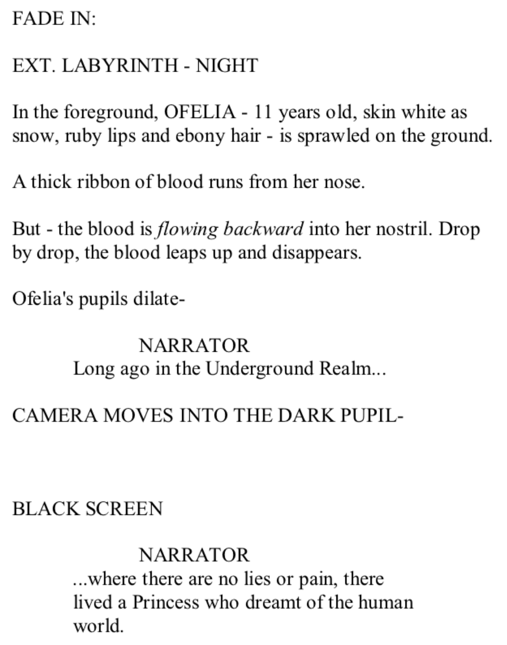
Literature review

Existing products and applications for AI image generation are limited. Available options include Stable Diffusion by Stability AI, Midjourney, and DALL-E by OpenAI. For this project, I evaluated the results produced by Stable Diffusion and DALL-E by asking each of them to produce various storyboards. Stable Diffusion consistently outperformed DALL-E at interpreting the prompts.

As background research, I also looked for products similar to my project. I found a prototype called Arter, which plans to integrate NLP methodologies and image association to generate storyboards. This concept differs from the project that I have outlined as its focus is on image recognition and generation of vector drawings and named entities, while mine is interested in evaluating the performance of existing AI tools (i.e. Stable Diffusion) given a script summarization.

Methodology/Dataset/Results

Screenplays are compositionally rigid narrative documents. Part of the first scene of Guillermo del Toro's *Pan's Labyrinth,* is pictured to the right. Screenplays may include transitions, scene headings, text, character headings, transitions, camera directions/shots, parentheticals, and sound cues. Additionally, auteurs — whose filmmaking style is distinct and the dominant influence over a production — may write in a more prosaic or poetic style. The lack of script-to-script standardization posed a challenge for dataset creation.



FADE IN:

EXT. LABYRINTH - NIGHT

In the foreground, OFELIA - 11 years old, skin white as snow, ruby lips and ebony hair - is sprawled on the ground.

A thick ribbon of blood runs from her nose.

But - the blood is *flowing backward* into her nostril. Drop by drop, the blood leaps up and disappears.

Ofelia's pupils dilate-

NARRATOR
Long ago in the Underground Realm...

CAMERA MOVES INTO THE DARK PUPIL-

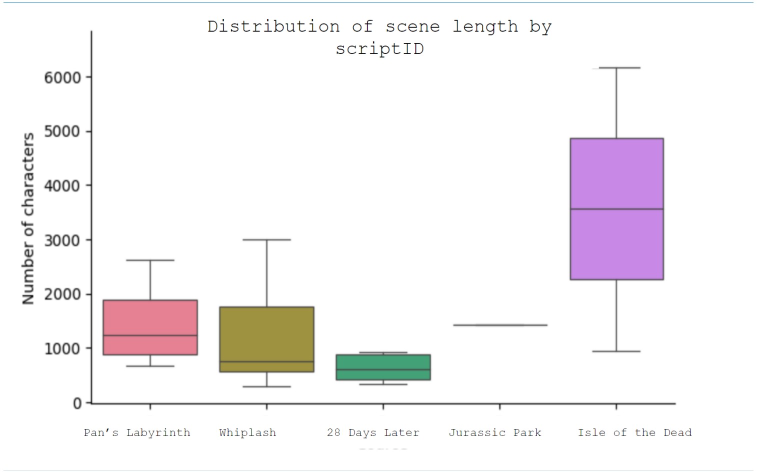
BLACK SCREEN

NARRATOR
...where there are no lies or pain, there lived a Princess who dreamt of the human world.

Fig. 1 First scene of *Pan's Labryinth*

To create the dataset for this project, I used manually annotated scripts from Kaggle. One limitation of the dataset is that the number of movies I could include is restricted to the manually annotated set.

After downloading the text files for *Pan's Labyrinth* (2006), *Whiplash* (2014), *28 Days Later* (2002), *Jurassic Park* (1993), *and Isle of the Dead* (1945), I used regular expressions to create a dataset split by *scriptID, sceneID,* and *text.* Then, I created a second dataset, which I ultimately used, which created an entry for each component in a scene. I wrote a function to clean the text by lowercasing all words and removing HTML tags and non-alphanumeric characters. After some testing, I found that removing these elements worsened the summaries generated. I also decided not to remove stopwords for this reason. The punctuation, formatting, and stopwords used provided some kind of signal to the model that informed its narrative structure.



The next challenge was in selecting a model for extractive summarization. I tested the long-t5-tglobal-base-16384 + BookSum model (1) and Longformer Encoder-Decoder (LED) for Narrative-Esque Long Text Summarization model (2). Both of these models were fine tuned on the BookSum dataset. The BookSum dataset — a collection of novels, plays, and stories — was created to improve tasks in longform narrative summarization. (1) is the long-T5 summarization model

developed by Google and fine tuned on the BookSum dataset. (2) is base model developed by AllenAI. The main difference between the models is in their attention mechanisms and predictive mechanisms — T5 is based on the prediction of masked words and LED is a probabilistic model.
Let us look at examples of summaries generated by each model on the opening scene of *Pan's Labyrinth* (*fig. 1):*

From (1): 'In this short scene, the action shifts to the nighttime of a young girl named Eglamour. She is eleven years old and has just had a stroke. Her nose is covered with blood and her hair is tied around her neck. Then she falls asleep. A long time ago, there was a princess living in the underground world who dreamed of the human realm. There was no pain or lies there lived if there were no lie or pain there existed a Princess who dreamt about the human world as well as the outside world. One day, the princess escapes from her keepers and disappears into the darkness.

From (2): 'In this short scene, the focus shifts to a young girl who has just escaped from her underground prison. She is shown as an 11-year old princess with a ribbon of red blood flowing down her nose and down her nostrils. The image captures Amelia\'s pupils dilating at the sight of the blood. It is clear that she is suffering from "cold sickness" and eventually death.'

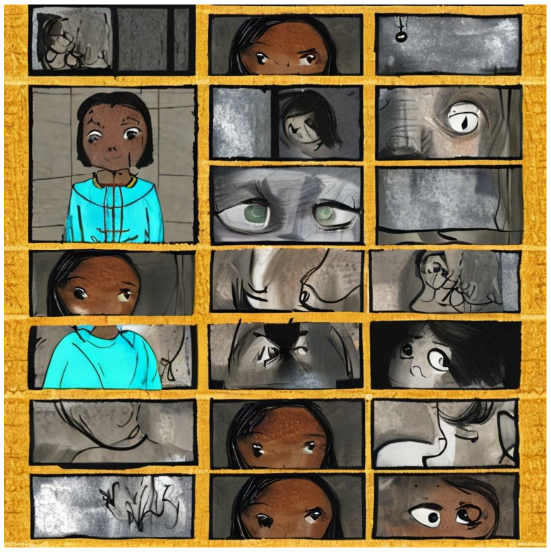


Fig. 2 Storyboard of the first scene of *Pan's Labryinth*

By manual review, I decided that (1) captured the magical realism of del Toro's original screenplay. As a test, I also generated a Stable Diffusion output from (1) (*fig. 2).*

One overall limitation for this project was that most of the models available on HuggingFace that were trained on narrative data worked best for long texts (BookSum). I had to decide between 1) Stringing together summarized scene components or 2) Summarizing whole scenes. Here is the output of this testing on the same scene.

(1) 'In this short scene, the narrator reveals that he has just returned from London with a letter from a friend. The letter is addressed to him as "Labyrinth," and it gives him an opportunity to express his disapproval at the way things are going in London.', 'In this short scene, Ofelia is eleven years old. She has a scarred nose due to a bout of scarlet fever. Her pupils dilate as she carries on.', ': Narrator : Narrog : : NAARATOR :NARATOR\': INTER : "NARRATOR / :Narroop : No : Niar : Men : In : Vol. : Ne : Ra : Non-N : Chapter : This : Nin : Also : The : National Narrators : Like : Nahr : Natratore : Name : Not : Unr n : At : Home : La : Nar Rank : Free : Underground : Here : Mar, : Even : You : Other : Nathan : Ap : News : Victor : Author : Actually : Remember : Yeah : Next : When : Okay : Rangor : U : Nu : New : Athletic : Welcome : All : Over : Intro : And : Where : We : Honor : Vi : Enter a : Short : History :', 'Long ago, in the underground Realm.... Long ago in that Underground Realm...... In the underground realm... Now re- a long ago in "the Underground Realmor... Long ago after in the

Unreal Realm...".', 'In this short scene, camera moves into the dark pony--into the "dark pil"..', ': Narrator : Narrog : : NAARATOR :NARATOR\': INTER : "NARRATOR / :Narroop : No : Niar : Men : In : Vol. : Ne : Ra : Non-N : Chapter : This : Nin : Also : The : National Narrators : Like : Nahr : Natratore : Name : Not : Unr n : At : Home : La : Nar Rank : Free : Underground : Here : Mar, : Even : You : Other : Nathan : Ap : News : Victor : Author : Actually : Remember : Yeah : Next : When : Okay : Rangor : U : Nu : New : Athletic : Welcome : All : Over : Intro : And : Where : We : Honor : Vi : Enter a : Short : History :'

(2) 'In this short scene, the action shifts to the nighttime of a young girl named Eglamour. She is eleven years old and has just had a stroke. Her nose is covered with blood and her hair is tied around her neck. Then she falls asleep. A long time ago, there was a princess living in the underground world who dreamed of the human realm. There was no pain or lies there lived if there were no lie or pain there existed a Princess who dreamt about the human world as well as the outside world. One day, the princess escapes from her keepers and disappears into the darkness.'

We can see that, when asked to summarize short scene components (like scene headings), the summarizer tried to abstract a summary when one wasn't necessary. For instance, it struggled with the character heading "NARRATOR" and outputted "Narrator : Narrog : : NAARATOR :NARATOR\': INTER : "NARRATOR / :Narroop : No : Niar : Men : In : Vol. : Ne : Ra : Non-N : Chapter : This : Nin : Also : The : National Narrators : Like : Nahr : Natratore : Name : Not."

For this reason, I decided to summarize the scenes intact. In general, I observed that the shorter the provided text, the more the model created an abstractive summary that was detached from the original screenplay.

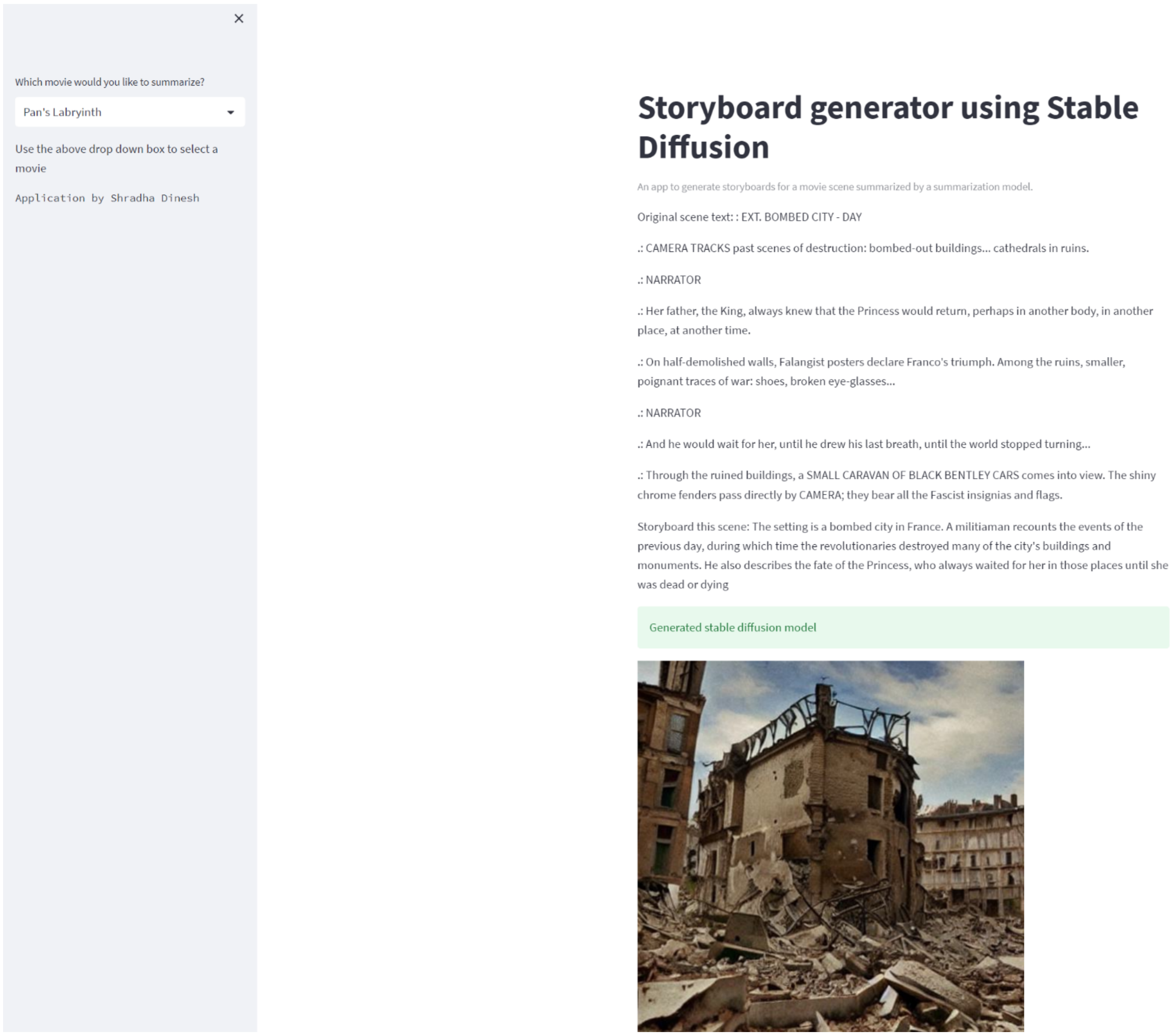
Results

To make the code for this project available, I created a Streamlit application (available on my GitHub).

NOTE: the available app takes 25-30 minutes to run without GPU due to the Stable Diffusion model. I have included code in my python file to provide the option to run on GPU. Instructions are on the ReadMe.md file. The code for pulling the Stable Diffusion model from HuggingFace was adapted from a img2text pipeline project by github user StatsGary.

I had difficulty incorporating the Stable Diffusion API directly, and opted for his pipeline method instead.

Here are two examples of the application output (*figs. 3 & 4)*:



Which movie would you like to summarize?

Pan's Labryinth

Use the above drop down box to select a movie

`Application by Shradha Dinesh`

Storyboard generator using Stable Diffusion

An app to generate storyboards for a movie scene summarized by a summarization model.

Original scene text: : EXT. BOMBED CITY - DAY

.: CAMERA TRACKS past scenes of destruction: bombed-out buildings... cathedrals in ruins.

.: NARRATOR

.: Her father, the King, always knew that the Princess would return, perhaps in another body, in another place, at another time.

.: On half-demolished walls, Falangist posters declare Franco's triumph. Among the ruins, smaller, poignant traces of war: shoes, broken eye-glasses...

.: NARRATOR

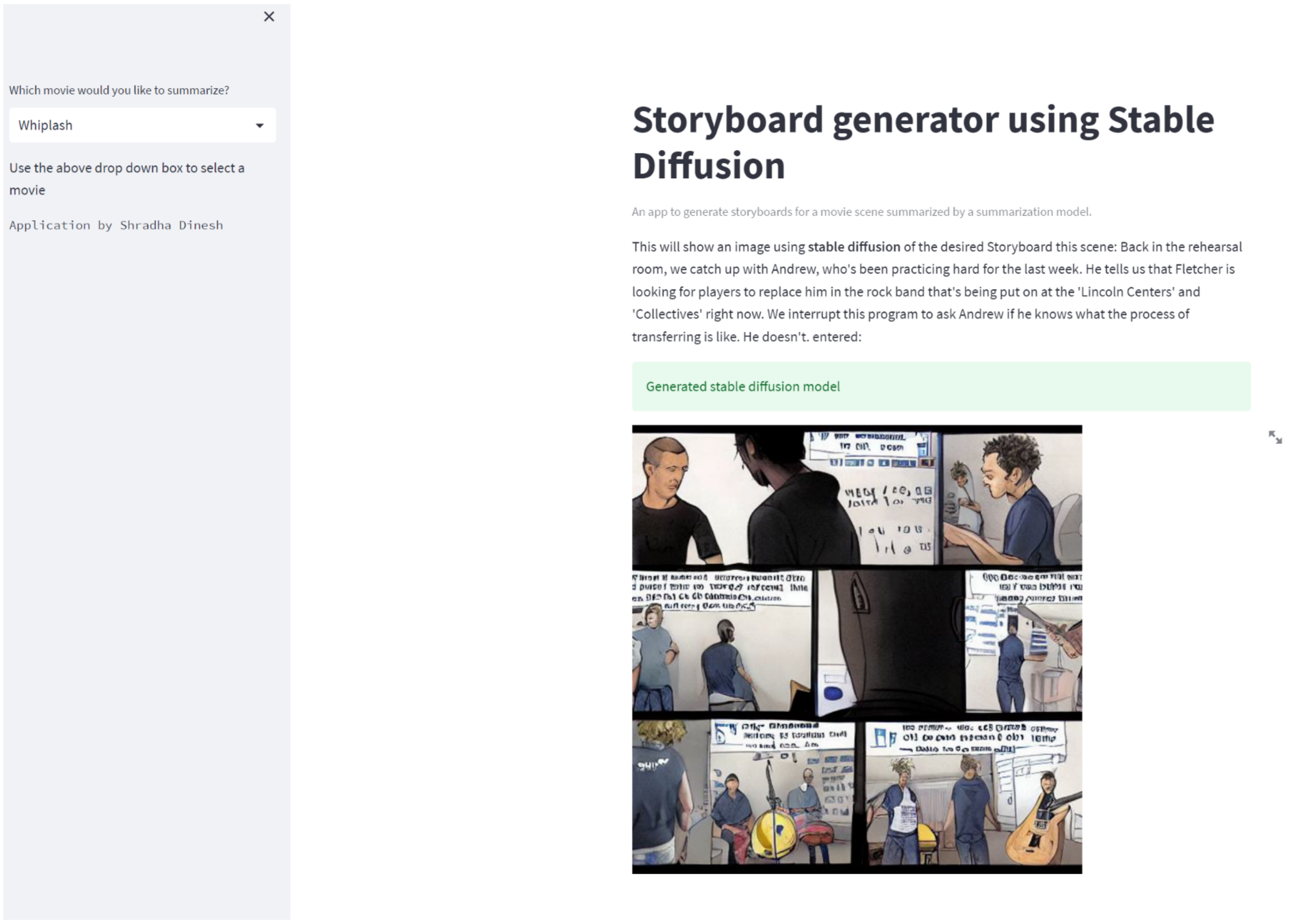
.: And he would wait for her, until he drew his last breath, until the world stopped turning...

.: Through the ruined buildings, a SMALL CARAVAN OF BLACK BENTLEY CARS comes into view. The shiny chrome fenders pass directly by CAMERA; they bear all the Fascist insignias and flags.

Storyboard this scene: The setting is a bombed city in France. A militiaman recounts the events of the previous day, during which time the revolutionaries destroyed many of the city's buildings and monuments. He also describes the fate of the Princess, who always waited for her in those places until she was dead or dying

Generated stable diffusion model

Fig. 3 Storyboard of a different scene from *Pan's Labryinth*


Which movie would you like to summarize?

Whiplash

Use the above drop down box to select a movie

`Application by Shradha Dinesh`

Storyboard generator using Stable Diffusion

An app to generate storyboards for a movie scene summarized by a summarization model.

This will show an image using **stable diffusion** of the desired Storyboard this scene: Back in the rehearsal room, we catch up with Andrew, who's been practicing hard for the last week. He tells us that Fletcher is looking for players to replace him in the rock band that's being put on at the 'Lincoln Centers' and 'Collectives' right now. We interrupt this program to ask Andrew if he knows what the process of transferring is like. He doesn't. entered:

Generated stable diffusion model

Fig. 4 Storyboard of a scene from *Whiplash*

Discussion

In light of the intent of the research — which was to test the ability of currently available AI models to summarize and illustrate a screenplay — we can see that there is a need for summarization models that specialize in short form narratives.

Notably, the BookSum dataset was trained on plays. When I prompted Stable Diffusion without specifically asking for a "screenplay of : ...," I would get a collage of images blocking out the actions of a play. I hypothesize that this happened because of BookSum's inclusion of plays. Play scripts, while similar, are different from film scripts. Transitions and camera directions, for example, are not present in plays.

An additional limitation of the summary models is that they are uncased, meaning that I could not ask the model to weight capitalized words with more importance. If this capability existed, I hypothesize that the summarization model would be stronger at recognizing named entities and preserving them in extractive summaries. This is a question for future research and model testing.

References

Beltagy, Iz, et al. Longformer: The Long-Document Transformer. 2, arXiv, 2020, doi:10.48550/ARXIV.2004.05150.

Guo, Mandy, et al. LongT5: Efficient Text-To-Text Transformer for Long Sequences. 2, arXiv, 2021, doi:10.48550/ARXIV.2112.07916.

Kryściński, Wojciech, et al. BookSum: A Collection of Datasets for Long-Form Narrative Summarization. 2, arXiv, 2021, doi:10.48550/ARXIV.2105.08209.

Szemraj, Peter. "Pszemraj/Long-T5-Tglobal-Base-16384-Book-Summary · Hugging Face." Pszemraj/Long-t5-Tglobal-Base-16384-Book-Summary · Hugging Face, HuggingFace, https://huggingface.co/pszemraj/long-t5-tglobal-base-16384-book-summary.

Appendices

StatsGary's Streamlit wrapper for Stable Diffusion:
https://github.com/StatsGary/stable-diffusion-streamlit/tree/main

Dataset Documentation (from GitHub ReadMe.md)

Title: movie_scenes_by_header.csv
Description: This dataset contains the scripts for 28 Days Later, Isle of the Dead, Jurassic Park, Pan's Labyrinth, and Whiplash.
Data Source: I used five manually encoded scripts from Kaggle for my analysis. I went with manual encoding to minimize the effect of machine learning miscodings on my overall project. Human mislabeling is also possible since this dataset was user created. I used regex statements to split headings and their respective text.
Date Created: March 10, 2023
Last Modified: April 20, 2023

Size: 85.6 KB — 6 columns, 543 entries
Format: Comma-separated values
Encoding: UTF-8

Columns: scriptID, sceneID, header, text
Column Descriptions:

scriptID: type (integer), unique integer identifier for each movie
sceneID: type (integer), unique integer identifier for each scene per movie

header: type (object), one of four scene headings (scene_heading, text, dialog, speaker_heading)
text: type (object), text that follows the header in the script
upper: type(object, list), extracted uppercase words from the screenplay text. Uppercase words in the screenplay indicate important characters, blockings, camera angles, etc.
tokens: type(object), preprocessed text from the text column
Title: movie_scenes_by_header.csv
Description: This dataset contains the scripts for 28 Days Later, Isle of the Dead, Jurassic Park, Pan's Labyrinth, and Whiplash.
Data Source: I used five manually encoded scripts from Kaggle for my analysis. I went with manual encoding to minimize the effect of machine learning miscodings on my overall project. Human mislabeling is also possible since this dataset was user created. I used regex statements to split headings and their respective text.
Date Created: March 10, 2023
Last Modified: April 20, 2023

Size: 85.6 KB — 6 columns, 543 entries
Format: Comma-separated values
Encoding: UTF-8

Columns: scriptID, sceneID, header, text
Column Descriptions:

scriptID: type (integer), unique integer identifier for each movie
sceneID: type (integer), unique integer identifier for each scene per movie
header: type (object), one of four scene headings (scene_heading, text, dialog, speaker_heading)
text: type (object), text that follows the header in the script
upper: type(object, list), extracted uppercase words from the screenplay text. Uppercase words in the screenplay indicate important characters, blockings, camera angles, etc.
tokens: type(object), preprocessed text from the text column
Title: movie_scenes.csv
Description: This dataset contains the scripts for 28 Days Later, Isle of the Dead, Jurassic Park, Pan's Labyrinth, and Whiplash.
Data Source: I used five manually encoded scripts from Kaggle for my analysis. I went with manual encoding to minimize the effect of machine learning miscodings on my overall project. Human mislabeling is also possible since this dataset was user created. Data was created by using regex statements to split scripts into a dataframe, and then the dataframes for all scripts were concatenated.
Date Created: March 10, 2023
Last Modified: April 20, 2023

Size: 50.6 KB — 4 columns, 34 entries

Format: Comma-separated values
Encoding: UTF-8

Columns: scriptID, sceneID, text, upper
Column Descriptions:

scriptID: type (integer), unique integer identifier for each movie
sceneID: type (integer), unique integer identifier for each scene per movie
text: type (object), the entire text of the scene, screenplay headings included
upper: type(object, list), extracted uppercase words from the screenplay text. Uppercase words in screenplays indicate important characters, blocking, camera angles, etc.