CQFPAS1

FYT Final Report

Automatic Movie Generation: Controllable Plot Generation and Video Retrieval

by

I-Tsun Cheng

CQFPAS1

Advised by

Prof. Pascale Fung

Prof. Qifeng Chen

Submitted in partial fulfillment

of the requirements for COMP 4981H

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2020-2021

Date of submission: April 12, 2021

Acknowledgement

First, I would like to thank my research supervisor, Professor Pascale Fung. Throughout the research journey, Pascale provided me with helpful advice and suggestions regarding the approaches utilized for the movie generation framework. As an undergraduate student who had limited research experience, Pascale offered me a valuable opportunity to work with her students in her lab. There, I learned so many things about research than I can ever imagine. Pascale also inspired me with her dedication and passion towards impactful research. Without her guidance and mentorship, this thesis would not have been possible.

I would also like to express my gratitude to Professor Qifeng Chen for his time being my research co-supervisor. I also very much appreciate Ziwei Ji for collaborating with me on related projects and discussing about natural language generation. As a graduate student at HKUST, she provided me with great insights into the field of research in Natural Language Processing. I am also grateful for all the lab friends who offered helpful feedback to the problems I encountered.

Lastly, I would like thank Noor Liza for her professional guidance on writing and structuring my reports. As my communication tutor, she gave me useful advice on the areas where I can improve and helped clarified many issues I had.

Abstract

Movies have taught us morals in life and inspired us in many ways possible. However, movies are usually expensive and costly to make. Therefore, we attempt to alleviate the complications involved with movie production by introducing a novel framework for movie generation. Our framework takes in a movie genre as input and generates a movie pertained to that genre as well as a plot that the movie attempts to follow. It consists of two primary components: 1) Controllable Plot Generation: generates the plot pertained to the genre, 2) Video Retrieval: retrieves the most relevant video clips from a database given a plot and combine them into a movie. For Controllable Plot Generation, we found that by using language models and specific attribute classifiers, our model is able to generate fluent and consistent plots that are strongly relevant with the input genre. For Video Retrieval, we built our custom algorithm that matches the plot with the video clips in the dataset by semantic similarity and abstractive summarization techniques. By different means of evaluation, we found that our movie generation framework is able to generate creative movies that is strongly relevant with the input genre and the generated plot.

Table of Contents

1 Introduction

1.1 Overview

For a long time, movie has provided us a great source of entertainment through its communication of ideas, stories, perceptions, and emotions. We watch movies on an occasional basis to be entertained and absorb new ideas. On a higher level, movies help shape our culture and reflect our thoughts of the world. Composed of moving images, they serve as a powerful medium of communication and education. A popular saying sums it all: “Movies dazzle us, entertain us, educate us, and delight us”.

Although movie has greatly benefited us and the society, the cost of movie production is immensely high. A movie production process typically involves screenwriting, casting, shooting, and editing and thus requires scriptwriters, actors and actresses, camera crew, and video editors, all of which make movie production tedious and costly.

To alleviate the human effort and time associated with movie production, an automatic movie generation software capable of generating videos according to certain inputs (e.g. genre, plot) can potentially help address the problem. Such a system can ease the workflow of production for film companies by generating high-quality and interesting videos, which can be directly used in the final production movie. Self-creators can also benefit from such a framework as they can utilize it to test and produce their own films quickly. The framework can also provide a source of inspiration for movie producers, which they can leverage to create more creative content.

While there has been a rising number of AI research on art, there is a lack of research on automatic movie generation. Existing works related to art focus on neural style transfer [1][2][3], poetry generation [4][5][6], story generation [7][8][9][10], music generation [11][12][13], etc. For those that are related to movies, they focus on script generation [14], automated trailer generation [15][16], movie retrieval ([17][18]). However, we found that there is no comprehensive work that discusses the construction of a movie generation framework that can generate movies according to some user inputs.

In this thesis, we propose an automatic movie generation framework that can generate short movies from scratch. Two of the most important elements that define a movie are genre and plot: genre tells the thematic category based on the depicted narration, aesthetics, or emotions presented in the film, while plot gives the overall progression of the events and actions conducted by characters. Therefore, to make our generated video more like a movie, our framework accepts a genre as the user input and generates a plot belonging to that genre in the intermediate process so that our final movie can attempt to condition on the genre and plot.

To generate a movie, we realized generating actual video frames from scratch is impractical. Modern research in video generation is still immature and requires expensive GPU hardware to reconstruct. Therefore, we decided to adopt retrieval-based methods to retrieve the relevant video clips from a dataset and combine them to create the final movie.

In general, our framework follows a two-step approach: 1) generate the plot pertained to a specific genre 2) retrieve the most relevant videos from a database based on the plot. To ease the construction of our framework, we divide it into two primary components: Controllable Plot Generation and Video Retrieval. In Controllable Plot Generation, we train a language model to generate movie plots and combine it with one or more attribute classifiers to guide the plot generation in the direction that more closely resembles our input genre. Then in Video Retrieval, we develop an iteration-based algorithm that takes a labelled database consisting of clip-subtitles pairs and matches a plot with the most relevant video clips by computing the semantic similarity between the plot and the subtitles. Abstract summarization is used at the end of each iteration to merge the plot and the subtitles of the retrieved video clips so that the modified plot becomes more relevant with the video contents. After a few iterations when the algorithm converges, we can finally concatenate all the video clips retrieved in the last iteration to form the final movie and use the last modified plot as its subtitles.

Our work provides two main contributions:

- To the best of our knowledge, we propose the **first controllable movie generation framework**. Our framework is able to take movie genre as user input and generate a movie of the corresponding genre.

- We show that our proposed framework can produce movies that are genre-relevant, plot-driven, and interesting based on automatic and human evaluation.

1.2 Objectives

The goal of this thesis is to build an automatic framework for movie generation. The output movie should show relevance to the genre specified by the user and also follow the plot generated in the intermediate process. To achieve our goal, we will mainly focus on the following objectives:

1. To gather and collect the datasets needed for plot generation and video retrieval for the movie generation framework.
2. To build a plot generation model capable of producing fluent movie plots pertained to different genres.
3. To develop a video retrieval algorithm that can retrieve the most relevant video clips in a dataset given a plot.
4. To generate interesting movies related to the input genre and plot by combining the plot generation model with the video retrieval algorithm.

1.3 Literature Survey

1.3.1 Text Generation

Plot generation is a specific application of text generation. Recent deep learning approaches for text generation have been proposed. Graves [18] proposed that generating or predicting sequential data like text with Recurrent Neural Networks (RNNs) [19] is effective. For better modeling capabilities, Long-Short Term Memory Networks (LSTMs) [20] was shown to generate fluent text by utilizing different gate mechanisms to modulate information for language modeling. Later, Vaswani [21] proposed the Transformer language model, which leverages self-attention layers to improve performance. It has proven to be a successful architecture for a wide variety of natural language tasks and has become the basis of many later works. As an extension from Vaswani's work, Radford [22] introduced GPT-2, a decoder-based transformer that achieved the state-of-the-art in many language modeling and generation tasks. The work suggests that fine-tuning a pretrained transformer like GPT-2 on a downstream task (e.g. language modeling) results in state-of-the-art performance. We leverage pretrained GPT-2 in our work for plot generation.

1.3.2 Controllable Text Generation

To make our plots controllable by a certain genre, we adopt controllable text generation, which aims to guide the generated text to a desired attribute. Current approaches for controllable text generation mainly fall in the three categories: fine-tuning models with Reinforcement Learning [23], training Generative Adversarial Networks [24], or training conditional language models [25][26]. The last approach has gathered more attention recently and has shown to be more effective. Keskar [27] proposed a conditional transformer language model, CTRL, that is trained on 50 different domains, specified with different control codes as input. However, CTRL is very computationally inefficient to fine-tune as it contains 1.6 billion parameters. To resolve this issue, Dathathri [28] introduced the Plug and Play Language Models (PPLM), which combines a pretrained language model and attribute classifiers to steer generation. PPLM requires no further training of the language model and allows any combination of attribute classifiers, making it very computationally efficient. Due to the lightweight advantage of PPLM, we adopt PPLM with our base GPT-2 language model to steer our plot generation to the desired genre.

1.3.3 Semantic Similarity and Sentence Embeddings

One of the main tasks in Video Retrieval is to compute semantic similarity to decide the video clips to match. The objective of semantic similarity is to calculate the similarity score between the semantic meanings of a pair of texts. In deep learning, a common approach towards semantic similarity is to encode the two texts with a certain embedding method and then compute their similarity using a similarity function, such as cosine similarity. As cosine similarity generally works better in measuring semantic meanings, the focus of this task primarily becomes finding the most effective sentence embedding method.

Sentence embeddings is a well-studied area that aims to map a sentence or document into a continuous vector representation that should encode their semantic meaning. A lot of methods for sentence embeddings have been proposed. Skip-Thought Vectors [29] trains an encoder-decoder model that tries to predict the surrounding sentences of a passage. InferSent [30] trains a siamese BiLSTM network with max-pooling on the labelled data of the Stanford Natural Language Inference dataset and the Multi-Genre dataset. They show that training on high-quality labelled dataset for sentence embeddings consistently outperforms unsupervised approaches like Skip-Thought Vectors. Universal Sentence Encoder [31] trains a transformer encoder that can encode sentences into embedding vectors, which can be applied to other tasks via transfer learning. Sentence-BERT [32] leverages a similar idea in that it uses a pretrained BERT and RoBERTa network and fine-tune it to produce useful sentence embeddings, resulting in SBERT and SRoBERTa. BERT [33] is a encoder-based transformer that pre-train deep bidirectional representations from text by jointing conditioning on both left and right context, and RoBERTa [34] is an optimized pretraining approach for BERT. Comparing with BERT/RoBERTa, SBERT/SRoBERTa reduces the time complexity of finding the most semantically similar pair of sentence in a corpus by a factor of around 46800 and outperforms other state-of-the-art approaches in semantic similarity. Since SRoBERTa slightly outperforms SBERT in the semantic similarity task, we use SRoBERTa for our main embedding approach to compute semantic similarity.

1.3.4 Abstractive Summarization

Summarization is the task of reducing the content of the text while preserving most of its meaning. According to how the content is selected or generated in the summary, summarization is generally categorized to extractive and abstractive techniques. Extractive summarization focuses on extracting the most salient phrases or sentences from the text and then rearranging them to form the summary. On the other hand, abstractive summarization generates new words and phrases that capture the meaning of the source text without simply extracting the key words from the source text. Recent works suggest that the attention has shifted from extractive to abstractive summarization as extractive has reached its peak performance, while abstractive has proven to be better at generating concise and coherent summaries. Hence, we focus on abstractive summarization.

Recent works on abstractive summarization adopt deep learning models to tackle the problem. Rush et. al. [35] used a feedforward neural network consisting of an attention-based encoder and beam-search decoder for sentence-level summarization. To improve upon their work, Chopra, Auli, and Rush [36] proposed to use a sequence-to-sequence model consisting of a conditional RNN to solve the problem. Many later works continued to use variants of the sequence-to-sequence model with the encoder-decoder architecture to map the input sequence to the output sequence. For instance, Nallapati, Zhou [37] used the bidirectional encoder with GRU-RNN and unidirectional decoder with GRU-RNN to model keywords and capture the hierarchical relationship between sentences and words more effectively. Jobson and Gutirrez [38] used an encoder-decoder RNN along with LSTM to generate the summaries and has found to work quite effective. Building upon the attention mechanism typically used in sequence-to-sequence models, Vaswani [21] proposed the Transformer, a successful architecture proven to be effective for various tasks, including abstractive summarization. It consists of deep encoder and decoder blocks containing self-attention layers to improve the performance. BART [39] was later proposed as a denoising autoencoder for pretraining sequence-to-sequence model such as Transformer-based architectures and was shown to achieve state-of-the-art results in abstractive summarization. We use BART for our abstractive summarization model.

2 Methodology

2.1 Design

2.1.1 Overall Framework

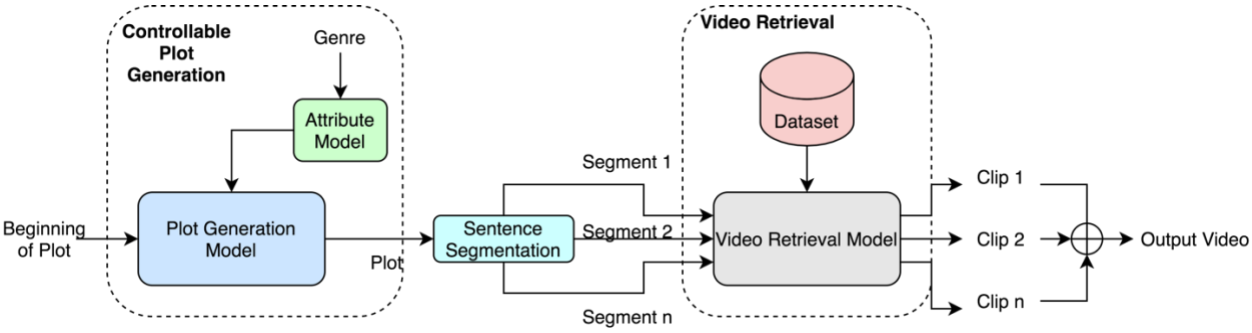

Figure 1: The Overall Framework of our Automatic Movie Generation System

Our task focuses on building an automatic movie generation framework that can produce movies relevant to the input genre and plot. The task is quite open-ended as it deals with both text and video. To simplify the problem, we divide our framework into two modular components, one for mainly generating text and another for selecting the most relevant videos, as shown in *Figure 1*. In particular, the Controllable Plot Generation is responsible for generating the plot while the Video Retrieval module for retrieving the best videos given the plot. We divide our system into these two modular components, since it simplifies the task into subtasks and there is meaningful interaction between the two components, as the generated plot in the Controllable Plot Generation provides a high-level planning for the movie in the Video Retrieval.

The Controllable Plot Generation module accepts genre and start tokens of the plot as the input and outputs the rest of the plot. Start tokens can be left empty so the module can generate the plot from scratch using only the genre as input. The module contains a trained plot generation model that can generate movie plots and an attribute classifier that can steer the generated plot in the style of the desired genre. After the plot is generated, we segment it into sentences via sentence segmentation. We then send each plot sentence to the Video Retrieval module one-by-one, which will retrieve the most relevant video clip from our

database using our proposed algorithm. Our video retrieval algorithm is iteration-based; during each iteration, it attempts to match a plot sentence with the best video clip by calculating the semantic similarity between the sentence and its textual labels (video description and subtitles). The subtitles of the video clip with the highest semantic similarity will then be merged together with the plot sentence via abstractive summarization to form a new plot sentence, which will be used to retrieve again in the next iteration. As will be shown in the experiments, the iteration normally converges within 5 iterations, and the video clips retrieved in the last iteration will be concatenated to form the final movie, while the modified plot will be used as its subtitles for the audience to follow.

In the Video Retrieval module, our proposed video retrieval algorithm changes the plot by performing abstractive summarization with the subtitles of the most relevant video clips during each iteration. Therefore, plot changes dynamically according to the video subtitles, and the one generated by the Controllable Plot Generation module only serves as the initial plot. We decided to change the plot dynamically since we realize doing so would increase the relevance of the plot with the video content (by merging plot with the subtitles) despite sacrificing plot consistency, as we find in the experiments later.

2.1.2 Controllable Plot Generation

The Controllable Plot Generation module contains a plot generation model which is able to generate a plot from scratch. Users can also specify the beginning of the plot by providing start tokens so that the generation is conditioned on them. For plot generation, we utilize pretrained GPT-2 previously trained on the *WebText* dataset, which contains 8 million web pages linked to by high-quality Reddit posts. We fine-tune the model on a plots corpus using the language modelling head to generate plots. Adopting the pretrained model leverages the implicit knowledge embedded within to ensure that the generated plots will be more fluent and informative.

We use GPT-2 for plot generation since it can generate long paragraphs of fluent text. The task of text generation can be primarily represented by language modelling, which is usually framed as an unsupervised distribution estimation of $p(x)$ where x is a sentence

composed of a sequence of tokens or symbols $(s_1, s_2, \ldots, s_n)$. Since language has a sequential ordering, we can calculate *p(x)* as the product of its conditional probabilities:

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, s_2, \ldots, s_{n-1})$$

By expressing the probability of a sentence in terms of the conditional probabilities of its tokens, models can focus on predicting the conditional probabilities and thus generate text token by token.

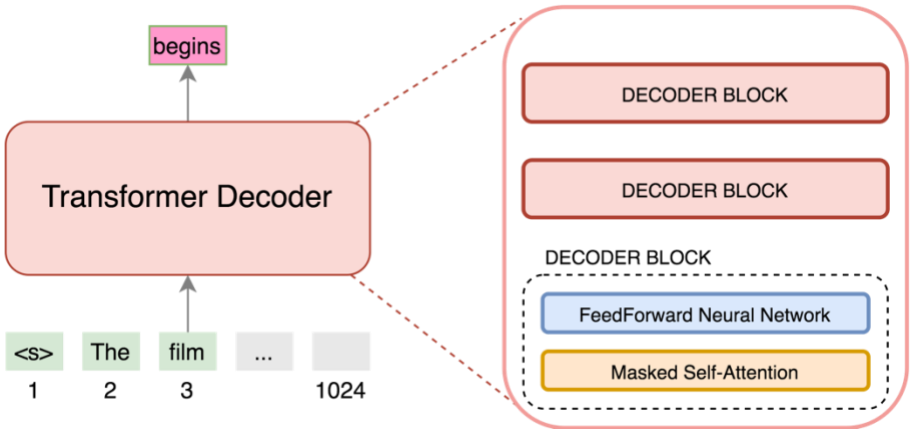

Figure 2: Transformer Decoder Architecture

GPT-2 is effective in generating text due to its architecture. GPT-2 [22] borrows the decoder from the original Transformer proposed by Vaswani [21], which is shown in *Figure 2*. The authors of GPT-2 show that when being trained on a huge corpus of text, it can model long-range dependencies very well and thus generate fluent text. The key mechanism behind GPT-2 is the masked self-attention mechanism that allows the next token to be generated to focus attention on the tokens already being generated. Self-attention scores the output token with each of the tokens already generated in the previous steps to decide the next generated token. To perform self-attention, GPT-2 uses three vectors to represent different entities. The Query vector is a representation of the current word to score against all other words, the Key vector is a representation of all of the generated words to match against the current word, and the Value vector is the actual word representations to add up to represent the current word after scoring how relevant each word is. In short, the self-attention matrix can be calculated as:

$$Attention\ (Q, K, V)\ =\ softmax\ (\frac{QK^T}{\sqrt{d_k}})V$$

where Q, K, V are resulting matrices from the concatenation of the Query, Key, and Value vectors.

To control the genre of the generated text, we adopt PPLM [28], which allows the generation to be controlled under a specific attribute (e.g. genre). Controllable generation models, such as PPLM, attempts to model probability *p(x|a)* where *a* is the desired attribute and *x* is the generated sample. It has been shown that $p(x|a) \propto p(a|x)\ p(x)$, and PPLM combines an attribute model that predicts *p(a|x)* and a base language model that predicts *p(x)* [27]. In our case, since we are controlling genre, we construct different attribute models for different genres using Bag-of-Words (BoW). For the BoW, we can build a list of words that frequently appear in a movie plot of the specific genre. For instance, for the romance genre, we can build a wordlist containing "love", "couple", "marriage", and for the crime genre, we can have "police", "detective", "murder", etc.

Given a set of keywords $\{w_1, \ldots, w_k\}$ pertained to a genre and the output distribution of the language model p_{t+1}, we can compute the log-likehood (LL):

$$\log p\ (a|x) = \log\left(\sum_i^k p_{t+1}(w_i)\right)$$

At every generation step *t*, PPLM updates the latent representation to generate the targeted words with higher probability [27]. PPLM shifts history H_t of the base transformer in the direction of the sum of two gradients: LL of a genre attribute *a* under *p(a|x)* and LL of the unmodified language model *p(x)*. Here, H_t is composed of key-value pairs generated up to time *t*. PPLM performs gradient update of H_t by:

$$\Delta \mathrm{H_t} \leftarrow \Delta \mathrm{H_t} + \alpha \frac{\nabla_{\Delta \mathrm{H_t}} \log \mathrm{p}\ (\mathrm{a}|\mathrm{H_t} + \Delta \mathrm{H_t})}{\left|\left|\nabla_{\Delta \mathrm{H_t}} \log \mathrm{p}\ (\mathrm{a}|\mathrm{H_t} + \Delta \mathrm{H_t})\right|\right|^{\gamma}}$$

where α is the step size, γ is the scaling coefficient for the normalizing term. After the latent representation is updated, the next token can be sampled. As a result of the previous latent update, the next token has a higher chance of being sampled from the BoW, and therefore the generated plot becomes more likely to be relevant to the input genre.

2.1.3 Video Retrieval

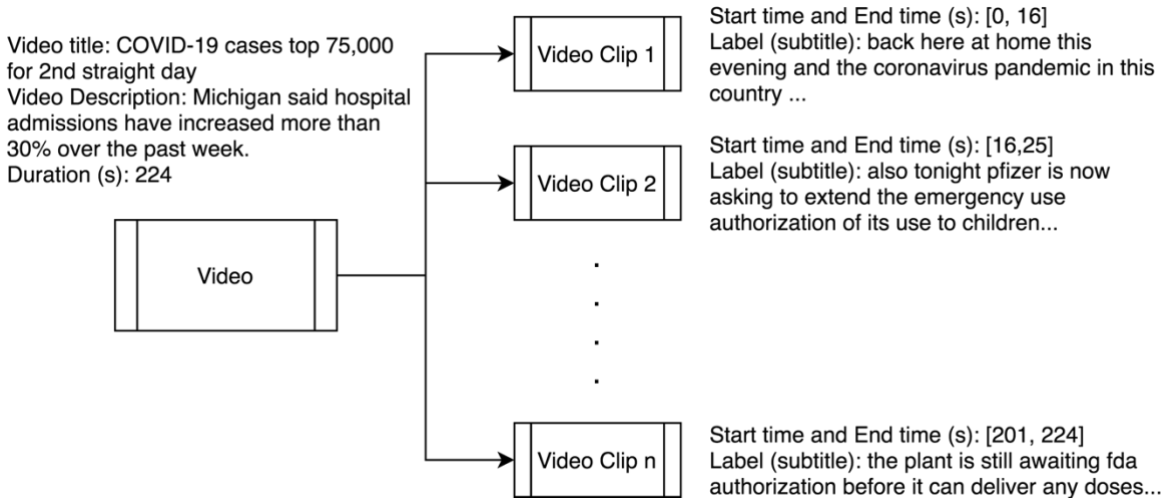

Figure 3: Video Dataset Format, example taken from a news video from ABC News

The Video Retrieval module consists of a text-to-video algorithm whose goal is to match each plot with the most relevant video clips. Before explaining the details of our video retrieval algorithm, we first discuss about the video database as the format is crucial to the algorithm. As shown in *Figure* 3, our video retrieval algorithm assumes the database contains the videos along with its metadata info, including video title, video description, and total duration of the video. In addition, it also assumes that each video can be split into different video clips, each containing the exact start time and end time as well as the corresponding labels that describe the content of the clip. As it can be difficult to get textual labels of video clips automatically, subtitles can be used as the labels.

For video retrieval, we do not use traditional video retrieval approaches that directly encode the textual features of the query and the visual features of the videos to compute similarity. In the traditional approaches, they do not have a labelled video dataset. However, in our case, we are able to gather textual information (video description, subtitles) for videos

in our dataset. Therefore, we can utilize the information for retrieval using semantic similarity, which will perform better than the traditional video retrieval approaches.

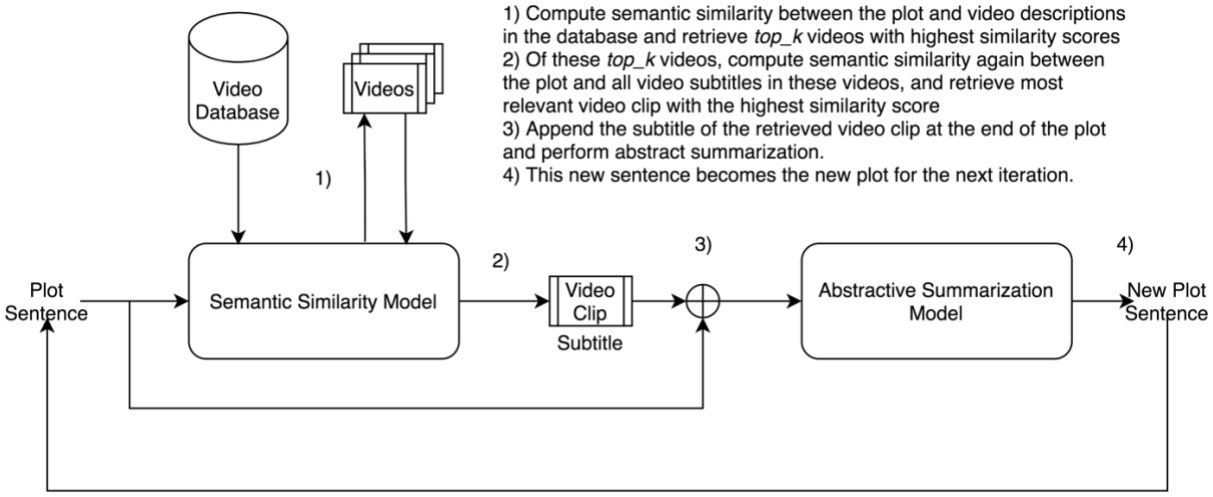

Figure 4: Our Video Retrieval Algorithm: retrieving the most relevant video clip from the dataset given a plot sentence

As shown earlier in *Figure 1*, our video retrieval algorithm takes in each plot sentence and attempts to find the best video clip for it. It does so by computing the semantic similarity using the semantic similarity model between the plot and the corresponding textual information associated with the videos. To make the plot more relevant with the video content, we perform the retrieval process iteratively and introduce the abstractive summarization model which can summarize the plot and the video subtitles at the end of each iteration, which can be used to retrieve again in the next iteration. *Figure 4* demonstrates the steps of our algorithm. Specifically, our algorithm works as follows:

For each iteration,

1) Given a plot sentence and a video database containing the videos and the information shown in *Figure 2*, group each video description in the database with the plot sentence to form plot-description pairs. For each pair, compute its pairwise semantic similarity using the semantic similarity model.

2) Find the *top_k* pairs that resulted in the highest semantic similarity scores. For each video description in the pair, find the corresponding video. For each video, find their corresponding video clips along with their subtitles.
3) We group each subtitle found in the previous step with the plot sentence to form plot-subtitle pairs. For each pair, compute its pairwise semantic similarity using the semantic similarity model.
4) Find the best pair that resulted in the highest semantic similarity score. The video clip associated with the subtitle in the best pair is assigned to the plot sentence in this iteration.
5) To retrieve again in the next iteration, append the subtitle in the best pair to the end of the plot sentence and perform abstract summarization by using the abstract summarization model. The new sentence generated will be used as the plot sentence for the next iteration.

As mentioned earlier, we chose to conduct several iterations and update the plot multiple times to increase the relevance between the plot and the video content. Without the iteration, the retrieved results will be heavily dependent on the videos contained in the video database. If there is limited number of videos in the database or if the video contents are all vastly different from the generated plot, then the retrieved results will be poor. Therefore, we decided to alter the plot dynamically with abstract summarization to strengthen the relevance between the plot and retrieved content.

In our algorithm, we have two models. The semantic similarity model is responsible for calculating the semantic similarity between the plot and videos for matching, while the abstractive summarization model is responsible for summarizing the plot and subtitles to form the new plot for the next iteration. For the semantic similarity model, we use RoBERTa [34], which is an improved pretraining approach for BERT. The modifications of RoBERTa on the original pretraining of BERT include:

1) Training the model for a longer period of time, with larger batches, and on more data
2) Removing the next sentence prediction objective
3) Training on longer sequences of data
4) Dynamically changing the masking pattern applied to the training data

These modifications lead RoBERTa to achieve state-of-the-art results in many downstream NLP tasks including GLUE, RACE, and SQuAD.

Although RoBERTa achieves state-of-the-art results in many downstream tasks, it causes a massive computational overhead in sentence regression tasks such as semantic similarity. In order to compute the semantic similarity between two texts, it requires both sentences to be fed into the network, which is computationally inefficient for large-scale semantic search like in our case. Therefore, SRoBERTa was proposed to reduce the overhead by using siamese and triplet network structures to derive semantically meaningful sentence embeddings. SRoBERTa was pretrained on SNLI + MultiNLI dataset, and we use the one that was fine-tuned on the Semantic Textual Similarity benchmark (STSb), a popular dataset for evaluating semantic similarity systems containing 8,628 sentence pairs from captions, news, and forums. During fine-tuning, the regression objective function used is the mean-square-error loss, which can be denoted by:

$$MSE = \frac{1}{N} \sum_{(s_a, s_b) \in P} (sim(s_a, s_b) - \widehat{sim(s_a, s_b)})^2$$

where *P* represents the set of all sentence pairs in the training dataset, *N* represents the total number of sentence pairs, $sim(u, v)$ represents the similarity score computed by the model between sentences *u* and *v*, and $\widehat{sim(u, v)}$ represents the actual similarity score in the dataset.

For abstractive summarization, we use BART [39], which is a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by 1) corrupting text with an arbitrary noising function, and 2) learning a sequence-to-sequence model to reconstruct the original text. It pre-trains a model combining bidirectional and auto-regressive transformers, and is found to achieve state-of-the-art results in summarization tasks. We use the pretrained BART that was previously trained on the CNN/Daily Mail News Dataset for abstractive summarization.

2.2 Implementation

2.2.1 Datasets

Genre	Number of Plots	Avg. plot length (tokens)
Action	2391	511.37
Comedy	7226	428.29
Crime	1605	412.6
Drama	9392	406.09
Horror	1579	617.87
Romance	2606	411.9
Sci-Fi	962	632.87
Western	973	346.75
Other	5399	485.1
Total Plots	**27809**	**445.15**

Table 1: Statistics for the Wikipedia Plots Corpus
Note: Total Plots do not add up because many movies belong to more than one genre

We fine-tune our GPT-2 model on the Wikipedia Plots Corpus[1] for plot generation. It contains a total of 27,809 movies and their corresponding info, including release year, title, origin, director, plot, genre, and wiki page. We downloaded the corpus from Kaggle and mainly make use of the plot and genre. We removed the movies that have genre "unknown".

[1] https://www.kaggle.com/jrobischon/wikipedia-movie-plots

Datasets	Number of Videos	Avg. video duration (secs)	Avg. number of segments per movie	Avg. segment duration (secs)	Avg. title length (tokens)	Avg. description length (tokens)	Avg. segment CC length (token)
ABC News	2873	404.978	27.602	14.671	10.356	31.468	40.972
Indian_Today	2518	376.888	39.833	9.461	15.458	51.397	24.548
Total	**5391**	**391.858**	**33.314**	**11.762**	**12.735**	**41.183**	**31.8**

Table 2: Statistics for our Custom Video Retrieval Dataset Consisting of News Videos from ABC News and Indian Today

For video retrieval, we built our own custom news dataset. We decided to use news videos as our video dataset because they are easier to find and provide valuable information on current events, which can potentially make our generated movie more up-to-date and relatable. Our news dataset contains 2873 most recent news videos from ABC News[2] and 2518 from Indian Today[3]. We downloaded the videos from their official channels on Youtube. Along with the videos, we scraped their metadata information, including their video titles and video descriptions. In addition, we scraped their subtitles, which are automatically generated by Youtube, as well as the start and end times associated with each subtitle. Since the subtitles automatically generated by Youtube are usually short sentence fragments and the duration of the corresponding video clips are typically short, we used a sentence tokenizer to combine several subtitles together into one to form a complete sentence. This effectively merges the corresponding video clips into one, with the new start time equal to the start time of the earliest video clip and the new end time equal to the end time of the latest video clip. By doing this preprocessing, each video clip will be longer, which will be more ideal for retrieval to produce the final movie. The sentence tokenizer we used is NLTK[4], a popular library for processing text.

[2] https://www.youtube.com/user/ABCNews
[3] https://www.youtube.com/channel/UCYPvAwZP8pZhSMW8qs7cVCw
[4] https://www.nltk.org/

2.2.2 Baselines

Controllable Plot Generation Baselines

We have the following baseline for plot generation to analyze genre control:

- GPT-2: We fine-tune GPT-2 on Wikipedia Plots Corpus and compare it against our model. We find the best hyperparameter settings for the baseline model to ensure fairness.

Video Retrieval Baselines

We compare our model to some semantic similarity baselines:

- DistilRoBERTa trained on paraphrase identification data (paraphrase-distilroberta-base-v1): Trained on large scale paraphrase data containing millions of paraphrase examples
- XLM-R trained on multi-lingual paraphrase identification data (paraphrase-xlm-r-multingual-v1): Trained on millions of multi-lingual paraphrase identification examples
- DistilBERT-base fine-tuned on STSb (stsb-distilbert-base)
- BERT-base fine-tuned on STSb (stsb-bert-base)
- BERT-large fine-tuned on STSb (stsb-bert-large)
- RoBERTa-base fine-tuned on STSb (stsb-roberta-base)

2.2.3 Choice of Genre

For plot generation, we selected three main genres to generate: Crime, Romance, and Science Fiction (Sci-Fi). We decided to select these particular genres because they are classic and easy to distinguish. The BoWs for the three genres that are used in our experiments are listed in the Appendix D. For video retrieval, only Crime and Romance are retrieved. This is because we are using news videos as our video database, and news videos hardly contain any sci-fi elements.

2.2.4 Controllable Plot Generation

We fine-tune GPT-2 on the Wikipedia Plots Corpus using Pytorch[5] and the Huggingface Transformers library[6], a library that provides thousands of state-of-the-art pretrained models for Natural Language Processing tasks. We used the GPT-2 medium variant that contains 24 layers, 1024 hidden dimensions, 16 heads for multi-headed attention, and 345 parameters. We trained the model on the corpus for 12 iterations, as we generally find the number to be large enough for the model to converge. To get the best hyperparameter setting, we tuned the model and evaluated it by calculating perplexity on the validation set. After tuning the model, we found the best hyperparameter set: learning rate 5e-5, label smoothed cross entropy loss function, and Adam optimizer (with epsilon 1e-8). For generation, we used top-k sampling decoder [40] where k is 10 and temperature is 1.0.

After fine-tuning, we constructed BoW for each genre, and ran PPLM on top of the fine-tuned GPT-2. An important parameter to test is the stepsize, which controls the degree of genre control. A higher stepsize usually results in stronger genre control but more repetition while a lower stepsize typically leads to weak genre control. After experimenting with different stepsizes, we found that using a stepsize of 0.04 for crime, 0.035 for romance, and 0.04 for sci-fi generally produce fluent and non-repetitive results with a noticeable number of words from the corresponding BoW list. In addition, we used 0.01 Kullback-Leibler Divergence scale [41] and 0.95 Geometric Mean scale [42] for all three genres. To give the model something to generate from, we provided the start tokens "*The film*" to let the model generate the rest of the plot.

2.2.5 Video Retrieval

In the video retrieval algorithm, we first retrieved *top_k* most relevant videos, which are then used to retrieve the most relevant video clip. We attempted several different values, *top_k=1, 2, 3, 4, 5*. We found that many values of *top_k* result in the same retrieved video clips for most plots, with *top_k=1* slightly outperforming the rest. Therefore, we decided to use *top_k=1*. We set our retrieval algorithm to run for 8 iterations, including the 1th iteration which performs retrieval using the initial plot generated by our PPLM.

[5] https://pytorch.org/
[6] https://huggingface.co/

In addition, there are a few design choices that we made in the video retrieval process. Before computing the semantic similarity between the plot and the video description, and between the plot and the subtitles, we first converted people's names intro their respective genders, if any. We did this so that difference in names will not affect the result of the similarity; instead, only the gender will. To convert names into genders, we downloaded and saved a list of male and female names so that they can be detected and converted. In addition, when trying to retrieve the most relevant video clip, we skipped those that have duration less than 7 seconds or more than 25 seconds to avoid retrieving a clip that is too short or too long.

To implement semantic similarity, we used the SentenceTransformers library[7], a Python framework for state-of-the-art sentence, text, and image embeddings introduced by the authors of Sentence-BERT [32]. SentenceTransformers provides many state-of-the-art pretrained transformer models for sentence embeddings, and we used the pretrained SRoBERTa that was fine-tuned on the STSb for the embedding approach. As mentioned before, we used cosine similarity as the similarity function to calculate the final similarity score.

For abstract summarization, we implemented BART using the Huggingface Transformers library, same with the GPT-2 for plot generation. We used the pretrained BART-large-cnn that was fine-tuned on the CNN/Daily Mail dataset. We set the minimum length of tokens to generate as 5, and the maximum length of tokens to 40 to prevent the summarized text to be too long. In addition, we set the decoder scheme of the summarization model to use beam search with a beam size of 4.

[7] https://www.sbert.net/

2.3 Testing

2.3.1 Controllable Plot Generation

To test plot generation, we primarily employed black box testing to examine the inputs and the corresponding outputs. We first tested our GPT-2 model to see if it can successfully generate plots by generating many samples. We inserted test cases with no start tokens and with start tokens to ensure that the model can generate a paragraph of text similar to the style of a movie plot in both cases. We also checked that the plots generated are fluent and non-repetitive, which gives the sign that our model is well-tuned.

After testing that our GPT-2 model can generate fluent plots, we tested the PPLM to ensure it can generate plots that condition well on the input genre. We inputted three different genres as test cases to see whether the generated plot includes some words listed in the corresponding genre BoW. We ensured that at least a few words in the generated plot are in the BoW so that it can reflect the genre.

2.3.2 Video Retrieval

For video retrieval, we also used black box testing to analyze the inputs and the produced results. The semantic similarity model is the key model behind our video retrieval algorithm, so it is important to ensure the model works correctly. We tested our semantic similarity model by first giving it sentence pairs to compute the semantic similarity. For example, we gave it a sentence pair consisting of two semantically similar sentences and another consisting of two nonrelevant sentences to calculate the similarity score. We expected the sentence pair consisting two similar sentences to achieve a higher score than the one consisting of two nonrelevant sentences. After testing the basic cases, we tested to see if our semantic similarity model can work well with our video database in video retrieval. We used our model to retrieve the top video clips by following our algorithm to see if the plot is similar to the video descriptions and the subtitles associated with the video clips. We expected them to have some semantic relevance and that the similarity score between them is not very low.

We also tested our abstractive summarization model. To test it, we gave the model sample test cases consisting of the plot and subtitle and analyzed the generated summary. We expected the summarization model to generate new phrases that are relevant to but not in the source texts, instead of extracting key words all the time. In addition, we also checked the videos generated in each iteration to ensure that each plot sentence matches with the video content of each video clip retrieved.

2.4 Evaluation

2.4.1 Evaluation Metrics

Controllable Plot Generation

We test the effectiveness of our plot generation approach by using a combination of automatic metrics and human evaluation. For plot, we mainly use **perplexity**, a common metric to evaluate the fluency of the generated texts from a language model. During the inference stage, we score the perplexity of our model against the baseline. The final perplexity score of the models is calculated by averaging the perplexities obtained from the three genres. In addition, we also use two other unsupervised metrics. **Distinct-n** gives the number of distinct n-grams divided by the total number of generated tokens, which measures the diversity of generated text. **Sentence length** is the average number of tokens in one sentence, which can reflect readability and complexity of the text.

Developing the best automatic metric is still a challenging problem in text generation. To complement with the automatic evaluation, we also conduct human evaluation. To evaluate genre control, we conducted A/B testing of our framework with the baseline. For our framework and the baseline, we generated 50 plots per genre (150 plots in total), and we asked the annotators on Appen[8] to choose which plot is more genre relevant. The annotators were given the following four choices: neither, plot 1 is more genre relevant, plot 2 is more genre relevant, or both are very genre relevant. To quantify the degree of genre control, we introduce a new metric genre control %, which represents the percentage of samples generated by a model that are more genre relevant than the other or very genre relevant. To

[8] https://appen.com/

improve the accuracy of our results, we decided to acquire 3 judgements per row of our evaluation data. To ensure the quality of the annotators, we included 10 sample test questions, of which each annotator must get 7 correct in order to qualify for the actual evaluation. In total, 271 annotators contributed to our evaluation.

Video Retrieval

To compare the performance of our semantic similarity model against the baselines, we set the **average similarity of the video title and description** of all videos in the database as the evaluation metric. The reasoning behind this evaluation metric is that in our semantic similarity task, we are retrieving a set of videos that yield the highest semantic similarity scores between their textual descriptions and the plot, which is similar to computing the similarity between the video titles and descriptions in our database. The plot and video titles are not exactly the same but hold common traits as they both describe the video content in the high level. Therefore, we can compare the performances of different models with ours by using them to calculate the average similarity between all video title and description associated with each video in our database. Better performing models should result in higher average similarity because the video title and description of most videos are closely related.

In addition, we evaluate our video retrieval algorithm by providing some unsupervised metrics on the retrieved results, including 1) **Average similarity score between the plot sentence and the description**, 2) **Average similarity between the plot sentence and the subtitles**, 3) **Average subtitles length**, and 4) **Average video clip duration**. The average similarity score between the plot sentence and the description refers to the average similarity between the plot sentences and the video descriptions of the videos containing the retrieved video clips, while the average similarity between the plot sentence and the subtitles refers to the average similarity between the plot sentences and the subtitles of the retrieved video clips. The average subtitles length refers to the average length of the subtitles from the retrieved video clips, calculated in tokens, and the average video clip duration refers to the average duration of the retrieved video clips, measured in seconds. For each of the 50 plots generated for crime and romance, we retrieve the best video clips for each iteration and evaluate them using these evaluation metrics.

2.4.2 Controllable Plot Generation

Model	Perplexity	Distinct-n (n=1,2,3)	Sentence Length
GPT-2	1.34	0.34/0.82/0.95	21.98
Ours	1.34	0.27/0.70/0.91	22.20
Human	-	0.07/0.44/0.79	20.03

Table 3: Automatic Evaluation for Controllable Plot Generation

For plot generation, we evaluate our model and the baseline on the test dataset using perplexity. As shown from *Table 3*, we find that applying PPLM to gain genre control does not occur at the expense of increased perplexity. In fact, using PPLM does not hurt the fluency of the generated plot, as the baseline GPT-2 and our model result in the same perplexity. In addition, we also find that our model leads to lower distinct n-grams than the baseline. This is expected and desired because lower distinct n-grams indicate that the generated plot does not contain too many distinct phrases, an outcome when several words from the BoW appear in the plot. Comparing our model to the baseline, the baseline generates many distinct n-grams and is therefore too random and lexically diverse, while our model's lower n-gram shows that it generates less randomly. The experiments show that our model generates more like a human-written plot than the baseline due to the lower number of distinct phrases found in a typical plot.

Model	Genre Control %
GPT2	28.67
Ours	56.67

Table 4: Human Evaluation of Genre Control for Controllable Plot Generation.

We also evaluate plot generation using human evaluation. As shown in *Table 4*, our framework achieves higher genre control % than the baseline, which demonstrates the effectiveness of our framework in genre control. Our model, which utilizes PPLM for genre control, is shown to be around twice as effective in generating plot specific to a genre than the baseline, which is not explicitly trained to control genre. This shows that PPLM performs much better in controlling the genre for plot generation than just simply using GPT-2, as it effectively generates words associated with a genre with higher probability.

2.4.2 Video Retrieval

Model	Avg. similarity score between video title and description
paraphrase-distilroberta-base-v1	0.3775
paraphrase-xlm-r-multilingual-v1	0.4130
stsb-distilbert-base	0.4349
stsb-bert-base	0.4398
stsb-bert-large	0.4482
stsb-roberta-base	0.4539
stsb-roberta-large	**0.4709**

Table 5: Automatic Evaluation for Semantic Similarity Baselines and Our Model

Table 5 shows the performances of our semantic similarity model with the other baselines. As the table shows, our model SRoBERTa-large fine-tuned on STSb, outperforms all other baselines in terms of the average similarity score of video title and description in our video dataset. We find that, in general, models that are trained on the STSb work better with our dataset in video retrieval than paraphrasing dataset. This is consistent with the intuition that models specifically trained for semantic similarity are more efficient than models trained for other tasks.

Among the models fine-tuned on the STSb, we find that SRoBERTa generally works better than SBERT for semantic similarity. This supports the fact that RoBERTa's optimized pretraining approach for BERT improves upon the one introduced in the original BERT. RoBERTa primarily improves upon BERT by training it for longer and over more data, while removing the next sentence prediction objective. In addition, our experiments show that SBERT and SDistilBERT result in around the same performances, with SBERT slightly outperforming SDistilBERT. Since the primary advantage of DistilBERT over BERT is that it is smaller, cheaper, and lighter, it is expected that DistilBERT experiences a little performance drop considering it being much smaller and efficient.

Moreover, we also discover that larger sizes of the same model are shown to work slightly better than their smaller counterpart due to its deeper architecture. Deeper

architectures usually have better performances since they contain more parameters to be trained and thus can learn better language representations. As demonstrated in the table, SRoBERTa-large and SBERT-large both outperform their lighter counterparts SRoBERTa-base and SBERT-base by a small margin.

The models fine-tuned on the paraphrase dataset do not appear to perform the best in semantic similarity. Between the two baselines trained on the paraphrase dataset, the XLM slightly has higher performance than the DistilRoBERTa. This can be explained by the fact that the former model is trained on multiple languages, while the latter is only trained on a single language. Multiple languages usually contain more language data and are more information-rich than a single language, making the XLM more effective than the DistilRoBERTa.

Iteration	Genre	Avg. similarity score with description	Avg. similarity score with cc	Average Segment CC length (token)	Avg. segment duration
1	Crime	0.4856	0.3856	30.3684	9.8684
	Romance	0.522	0.4307	31.8787	11.0606
	Average	**0.5038**	**0.40815**	**31.12355**	**10.4645**
2	Crime	0.5416	0.5529	33.2894	11.2894
	Romance	0.5143	0.5552	33.3636	11.8484
	Average	**0.52795**	**0.55405**	**33.3265**	**11.5689**
3	Crime	0.5531	0.5609	33.9473	11.421
	Romance	0.5442	0.6117	31.4545	10.6666
	Average	**0.54865**	**0.5863**	**32.7009**	**11.0438**
4	Crime	0.5539	0.5676	34.3157	11.5789
	Romance	0.5421	0.6014	31.4242	10.5151
	Average	**0.548**	**0.5845**	**32.86995**	**11.047**
5	Crime	0.5534	0.5635	34.3684	11.7368
	Romance	0.5424	0.6003	31.4242	10.5151
	Average	**0.5479**	**0.5819**	**32.8963**	**11.12595**
6	Crime	0.5569	0.5634	34.1578	11.6578
	Romance	0.541	0.5969	31.4242	10.5151
	Average	**0.54895**	**0.58015**	**32.791**	**11.08645**

7	Crime	0.5587	0.56	34.3947	11.6578
	Romance	0.543	0.5872	32.8484	10.7878
	Average	**0.55085**	**0.5736**	**33.62155**	**11.2228**
8	Crime	0.5587	0.56	34.3947	11.6578
	Romance	0.556	0.5997	32.8484	10.7878
	Average	**0.55735**	**0.57985**	**33.62155**	**11.2228**

Table 6: Automatic Evaluation for Our Video Retrieval Algorithm for each Iteration

Table 6 shows the evaluation results of our video retrieval algorithm. Our experiments show that in the 1st iteration, the initial plot has a relatively low semantic similarity score of 0.408 with the subtitles of the matched video clip. This suggests that without changing the plot, the content of the corresponding retrieved video clip is not very relevant with the plot. By updating the plot through abstractive summarization with the subtitle, we can see that the relevance becomes stronger. In particular, after the 2nd iteration, we can see that the average similarity score between the plot and the subtitle increases substantially to 0.554, which indicates that the new updated plot now becomes more similar with the subtitles of the new retrieved video clip in the current iteration and therefore also the video content of the corresponding clip. This increase in average similarity is caused by the abstractive summarization that happens at the end of each iteration, which merges the plot with the subtitles so that the new plot becomes closer with the subtitles in their joint vector space. In addition, we can also see that the average similarity between the plot and the video description also increases indirectly even if the plot is not merged with the description, demonstrating the effectiveness of our summarization approach between the plot and the subtitle.

Our experiments show that the most substantial improvements occur between the 1th, 2nd iteration and the 2nd, 3rd iteration. After the 3rd iteration, the algorithm does not have any significant improvements in terms of the average similarity with description and the average similarity with subtitle. The reason behind is that within the first three iterations, the syntax and style of the plot might be significantly different from the subtitle. Therefore, updating the plot according to subtitle will change the plot more towards the subtitle in the first few iterations, and the video clips matched differ quite frequently in each of the earlier iterations.

However, in the later iterations, the plot has already experienced many updates and is therefore more semantically similar to the subtitle. Thus, the plot might no longer be updated when doing abstractive summarization and the video clips matched might remain the same in the later iterations. We generally see that the video retrieval algorithm converges within 5 iterations, after which there is hardly any difference in the video clips retrieved.

3 Discussion

3.1 Controllable Plot Generation

Example	Genre	Generated Plot
1	Crime	The film opens with the narration: When an orphaned child is found dead, police investigate the death and find that it is not murder as they are led to believe. The investigating officer, who works for the local gangster boss Jocko, is suspected of being the killer. He tries to convince a reluctant Inspector Chiu-san to take the case and put the gang's reputation on the line, but he refuses and is sentenced to jail. The only witness who can prove his innocence is Chiu-San's brother, Captain Ma, who has a romantic interest in Chiu-San. When Jocko is released from prison, he finds his men abduct him. Inspector Chiu-San is on the hunt, but is unable to stop the criminal. Inspector Chiu-san is involved in a car chase while trying to rescue Ma's young brother. When the car he was riding in crashes into another, Inspector Chiu-San, his wife, and their newborn child are hurt.
2	Romance	The film begins with a flashback to a childhood in a remote village called Gomma. The villagers are poor and there seems to be nothing better than a fair and luxurious life. One day a young boy, Ramesh comes to stay with the family. He falls in love with Gomma beauty, Parvathi and falls in love with her. The love blossoms between Ramesh and Parvathi, even though it is not love at first sight. One day when Parvathi and Ramesh were dancing at the house, her father got a shock when he see Ramesh as a good human being. He decides to take a liking to that girl.
3	Sci-Fi	The narration of a dying scientist explaining that the human mind is a vessel for the creation of the universe; the brain is the vessel, and is thought to be the center. John Galt and James Stewart star ships the spaceship the gila with a mission. Galt wants to travel to mars with his spaceship, while Stewart is to go to mars to study a chimpanzee he has been studying. They arrive in earth space on the murn, and are greeted by the commander James Stewart and lieutenant colonel John Hargan, both of their respective mars missions. Galt, the only earth space ship, is an

		advanced space vehicle that has been in orbit since the planet mercury. The ship is commanded by dr. john l. rell, a scientist who has been studying earth space for two years. His wife martha has gone into a deep depression because of a space suit malfunction. He works at the laboratory to space station alpha, a space ship with three martian space suits.

Table 7: Generated plot samples for Crime, Romance, and Sci-Fi

Table 7 shows the three different plot samples that are conditioned on each of the three genres, generated from our model. As shown from Table 1, the plots generated by our model is pretty fluent and consistent. Although there might be some grammar mistakes, the generated texts are smooth to read and does not frequently disrupt the flow of the sentence. The plots are also consistent in that the same character reappears multiple times within the plot instead of a single time. Fluency and consistency are largely accredited to GPT-2's large pretrained dataset and effective transformer-based decoder. GPT-2 was pretrained on a huge dataset consisting of millions of webpages, providing a large source for the model to learn good language representations. The Transformer-based architecture allows GPT-2 to model long-range dependencies using its multi-headed self-attention mechanism and deep decoder layers, enabling it to generate paragraphs of fluent and consistent text.

In addition, we can see that all three plots belong to their associated genre as they contain some of the words that highly appear in the contexts of the genres. For instance, we can see that the crime plot contains the words "dead", "police", "murder", "officer", "killer", "jail", "criminal"; the romance plot contains the words "love", dancing", and "liking"; and the sci-fi plot contains the words "scientist", "universe", "spaceship", "mars", "earth", "space", "orbit", "space suit", etc. These words are strongly related with their associated genres. The inclusion of these words in the generated plot shows PPLM's effectiveness of steering the style toward specific genres.

Although the generated plots are fluent, consistent, and pertained to their associated genre, they lack story development and common sense. The generated plots do not show a clear logical progression of events and conflicts and instead sometimes seem to generate nonrelevant text. They also contain some sentences that do not make sense. However, the

plots generated by our model still capture the style of a typical plot, as they contain character names and action, two important elements that distinguish a plot from a paragraph of text.

3.2 Video Retrieval

Plot Sentence	Iteration Number					
	1 (Original)		2		3	
	Plot	Matched CC	Plot	Matched CC	Plot	Matched CC
1	when jocko is released from prison, he finds his men abduct him.	36arnhardt36 kidnapping case has come to an astonishing end police believe they have the victim who was snatched from the street 18 years ago as a young girl.	When jocko is released from prison, he finds his men abduct him . Police believe they have the victim who was snatched from the street 18 years ago	36arnhardt36 kidnapping case has come to an astonishing end police believe they have the victim who was snatched from the street 18 years ago as a young girl.	When jocko is released from prison, he finds his men abduct him . Police believe they have the victim who was snatched from the street 18 years ago.	36arnhardt36 kidnapping case has come to an astonishing end police believe they have the victim who was snatched from the street 18 years ago as a young girl.
2	when the car he was riding in crashes into another, inspector chiu-san, his wife, and their newborn child are hurt.	Nascar legend dale 36arnhardt jr. his wife and their baby daughter there seems to be an airplane they landed completely.	Nascar legend dale 36arnhardt jr. crashes into another car and his wife and their newborn child . The crash is the result of a collision between the two cars .	nascar legend dale 36arnhardt jr. his wife and their baby daughter there seems to be an airplane they landed completely.	Nascar legend dale 36arnhardt jr. crashes into another car and his wife and their newborn child . The crash is the result of a collision between the two cars .	nascar legend dale 36arnhardt jr. his wife and their baby daughter there seems to be an airplane they landed completely.

Table 8: Matched result Crime samples by our Video Retrieval Algorithm for each Iteration (The parts highlighted in red indicate the updated sentences from the previous iteration)

Plot Sentences	Iteration Number					
	1 (Original)		2		3	
	Plot	Matched CC	Plot	Matched CC	Plot	Matched CC
1	He falls love with	i feeling her mom hugged	He falls in love with	i feeling her mom hugged me and just	He falls in love with	i feeling her mom hugged me and just

	Gomma beauty, Parvathi and falls love with her.	me and just staring into her eyes to some it's a strange love but they had to give it a shot when you were young and in love and follow your heart.	Gomma beauty, Parvathi and falls love with her . i feeling her mom hugged me and just staring into her eyes to some it's a strange	staring into her eyes to some it's a strange love but they had to give it a shot when you were young and in love and follow your heart.	Gomma beauty, Parvathi and falls love with her . i feeling her mom hugged me and just staring into her eyes to some it's a strange	staring into her eyes to some it's a strange love but they had to give it a shot when you were young and in love and follow your heart.
2	He decides to take a liking to that girl.	subscribe that they say they fell in love almost immediately instant i have it vividly in my head just looking over.	i have it vividly in my head just looking over. He decides to take a liking to that girl.	much does he love her let george clooney count the ways in an interview with entertainment tonight the actor just couldn't stop gushing about his wife of eight months a mall listing the reasons he fell in love with her from her great sense of humor to her eccentric but fun sense of style but one thing a mall doesn't love about george is his tv.	George Clooney couldn't stop gushing about his wife of eight months a mall .	much does he love her let george clooney count the ways in an interview with entertainment tonight the actor just couldn't stop gushing about his wife of eight months a mall listing the reasons he fell in love with her from her great sense of humor to her eccentric but fun sense of style but one thing a mall doesn't love about george is his tv.

Table 9: Matched result Romance samples by our Video Retrieval Algorithm for each Iteration

From the matching results displayed in *Table 8* and *Table 9*, we can see that the subtitles of the matched video clip show some relevance with the plot. Although the subtitle retrieved might not always mean exactly the same as the plot, they usually have a similar action or event happening. For instance, for sentence 1 during the 1st iteration in *Table 8*, the crime plot indicates that a character is abducted by a group of people, and the subtitle depicts a similar situation as it mentions a “kidnapping case”. We can also see the same thing in the romance plot. For sentence 1 during the 1st iteration in *Table 9*, the romance plot illustrates that a man falls in love with a girl, which is relevant with the “strange love” and “they had to give it a shot since they are young and in love” as indicated in the subtitle. These examples show that our semantic similarity model is quite effective in retrieving semantically relevant video clips.

Although most of the plot show relevance with the retrieved subtitles, there might be cases where the plot is hardly relevant with the subtitles. This might happen when the content of the plot is vastly different from the content of dataset used for video retrieval. This is an inherent limitation of any retrieval-based approach whose results are dependent on the video dataset.

In addition, our experiments show that our summarization model attempts to merge the plot and the retrieved subtitles to update the plot for the next iteration. In many cases, the model directly extracts the key information from the two source texts and combine them to form the summary. We find that this might be because the two texts might be difficult to summarize since they might be discussing about different things, making the model to summarize by directly combining key phrases from both source texts. This is caused mainly by the inherent limitation of the video dataset as discussed previously. In other cases, the summarization model performs quite well by trying to synthesize the two. For instance, for sentence 2 during the 1st iteration in *Table 8*, the crime plot indicates that a car crashes into another car, while the subtitles discusses about the person "nascar legend dale earnhardt jr". The resulting summary directly substitutes the person as the driver of the car that resulted in the collision, a smart decision made by the summarization model. In addition, the model adds new information not presented in the plot and subtitle by mentioning " the crash is the result of a collision between the two cars". This indicates that the model does not simply extract key phrases and rearrange them as in extractive summarization when the two source texts are similar enough to be summarized well.

3.3 Overall Framework

In general, the movie generated by our framework is quite creative and interesting. We generated two short movies, one belonging to the crime and the other belonging to the romance genre, and uploaded them online[9]. As the subtitles of the generated movie is the plot of the last iteration during the video retrieval process, the creativity of the movie comes from the creativity of the plot generated by our GPT-2 and PPLM model.

[9] https://drive.google.com/drive/folders/1bugeu8gPlCIfNd7MH1ipT7Ko2Xt1HgP3?usp=sharing

Figure 5: A Video Frame taken from a Generated Crime Movie

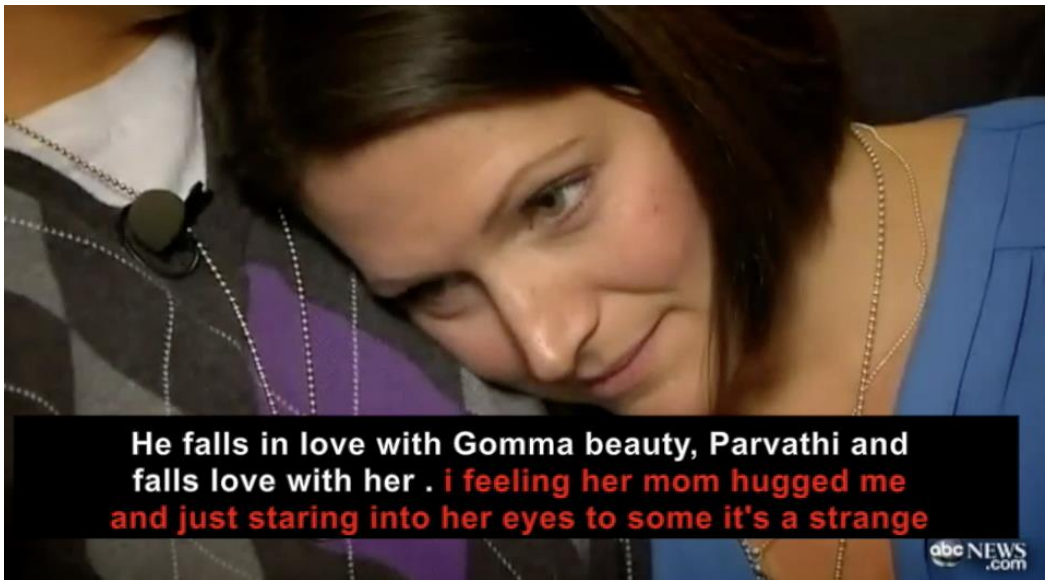

Figure 6: A Video Frame taken from a Generated Romance Movie

In addition, we can see that many video clips in the generated movie are relevant with the plot, inserted as subtitles. For instance, as depicted in *Figure 5*, the car crash described in the plot is reflected by the heavy smoke occurring on a road in the video. From *Figure 6*, we can see that a female figure is hugging a person and staring at something in the video as illustrated in the plot. This demonstrates the effectiveness of our framework in retrieving relevant video clips in our database.

Figure 7: Another Video Frame taken from a Generated Romance Movie

However, a drawback of our framework is the inability to match to the correct video given a named entity. Specifically, if our plot contains a named entity like a person's name, then no mechanism is enforced in our framework to guarantee that the retrieved video is the right one containing the person. For instance, in *Figure 7*, the plot contains names including "Vice President Pence" and "Trump". However, the matched video clip contains President Barack Obama. This happens because our framework is not specially designed to perform named entity recognition and thus is a limitation of our framework.

4 Conclusion

In this thesis, we focused on building a movie generation framework capable of generating movies that show elements of a specific genre and follow a plot. We showed that our controllable plot generation model consisting of GPT-2 and PPLM is able to generate fluent and consistent plot that is relevant with the input genre. In addition, we also demonstrated that our video retrieval algorithm containing a SRoBERTa semantic similarity model and BART abstract summarization model performs quite well in retrieving the most relevant video clips from our video dataset. Although our framework can generate movies quite robustly, we believe there are improvements to be made. Future areas of research can focus on allowing more granular control of the movie generation like specific events or characters, adding music to the movie to make it more interesting, or improving the video retrieval algorithm such as adding named entity recognition.

5 References

[1] Kotovenko, D.; Sanakoyeu, A.; Ma, P.; Lang, S.; and Ommer, B. 2019. A content transformation block for image style transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 10032–10041.

[2] Z. Xu, M. Wilber, C. Fang, A. Hertzmann, and H. Jin, "Learning from multi-domain artistic images for arbitrary style transfer," arXiv preprint arXiv:1805.09987, 2018.

[3] Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2018. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv: 1808.04537.*

[4] J. Benhardt, P. Hase, L. Zhu, and C. Rudin, "Shall i compare thee to a machine- written sonnet? an approach to algorithmic sonnet generation," arXiv preprint arXiv:1811.05067, 2018.

[5] Z. Liu, Z. Fu, J. Cao, G. de Melo, Y.-C. Tam, C. Niu, and J. Zhou, "Rhetorically controlled encoder-decoder for modern chinese poetry generation," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1992–2001.

[6] B. Liu, J. Fu, M. P. Kato, and M. Yoshikawa, "Beyond narrative description: Gen- erating poetry from images by multi-adversarial training," in Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 783–791.

[7] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," arXiv preprint arXiv:1805.04833, 2018.

[8] J. Guan, F. Huang, Z. Zhao, X. Zhu, and M. Huang, "A knowledge-enhanced pre- training model for commonsense story generation," Transactions of the Association for Computational Linguistics, vol. 8, pp. 93–108, 2020.

[9] L. Yao, N. Peng, R. Weischedel, K. Knight, D. Zhao, and R. Yan, “Plan-and-write: Towards better automatic storytelling,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 7378–7385.

[10] H. H. Mao, B. P. Majumder, J. McAuley, and G. W. Cottrell, “Improving neural story generation by targeted common sense grounding,” arXiv preprint arXiv:1908.09451, 2019.

[11] H.-W. Dong and Y.-H. Yang, “Convolutional generative adversarial networks with binary neurons for polyphonic music generation,” arXiv preprint arXiv:1804.09399, 2018.

[12] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, “Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment,” arXiv preprint arXiv:1709.06298, 2017.

[13] S. Vasquez and M. Lewis, “Melnet: A generative model for audio in the frequency domain,” arXiv preprint arXiv:1906.01083, 2019.

[14] Y. Zhu, R. Song, Z. Dou, J.-Y. Nie, and J. Zhou, “Scriptwriter: Narrative-guided script generation,” arXiv preprint arXiv:2005.10331, 2020.

[15] G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa, “Automatic trailer generation,” in Proceedings of the 18th ACM international conference on Multimedia, 2010, pp. 839–842.

[16] M. Hesham, B. Hani, N. Fouad, and E. Amer, “Smart trailer: Automatic generation of movie trailer using only subtitles,” in 2018 First International Workshop on Deep and Representation Learning (IWDRL). IEEE, 2018, pp. 26–30.

[17] Y. Deldjoo, M. G. Constantin, B. Ionescu, M. Schedl, and P. Cremonesi, “Mmtf-14k: a multifaceted movie trailer feature dataset for recommendation and retrieval,” in Proceedings of the 9th ACM Multimedia Systems Conference, 2018, pp. 450–455.

[18] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele, “Movie description,” International Journal of Computer Vision, vol. 123, no. 1, pp. 94–120, 2017.

[19] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850 (2013).

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] A.Vaswani,N.Shazeer,N.Parmar,J.Uszkoreit,L.Jones,A.N.Gomez,L□.Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998–6008.

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI Blog, vol. 1, no. 8, p. 9, 2019.

[23] Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, 722 A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine- 723 tuning language models from human preferences. *arXiv 724 preprint arXiv:1909.08593.*

[24] Yu, L.; Zhang, W.; Wang, J.; and SeqGAN, Y. Y. 2016. Se- 702 quence Generative Adversarial Nets with Policy Gradient. 703 arXiv e-prints, page. *arXiv preprint arXiv:1609.05473.*

[25] Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; and Okumura, M. 2016. Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552.*

[26] Ficler, J.; and Goldberg, Y. 2017. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633.*

[27] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, “Ctrl: A conditional transformer language model for controllable generation,” arXiv preprint arXiv:1909.05858, 2019.

[28] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, “Plug and play language models: a simple approach to controlled text gen- eration,” arXiv preprint arXiv:1912.02164, 2019.

[29] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 3294–3302. Curran Associates, Inc

[30] Alexis Conneau, Douwe Kiela, Holger Schwenk, Lo¨ıc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics

[31] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. arXiv preprint arXiv:1803.11175.

[32] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[33] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[34] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[35] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[36] Chopra, Sumit, Michael Auli, and Alexander M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

[37] Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

[38] Jobson, Elliott, and Abiel Gutiérrez. "Abstractive Text Summarization Using Attentive Sequence-To-Sequence RNNs." (2016): 8.

[39] Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

[40] Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

[41] Van Erven, T., & Harremos, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, *60*(7), 3797-3820.

[42] Crawford, G. B. (1987). The geometric mean procedure for estimating the scale of a judgement matrix. *Mathematical Modelling*, *9*(3-5), 327-334.

6 Appendix

6.1 Appendix A: Meeting Minutes

6.1.1 Minutes of the 1st Project Meeting

Date: Sep 1, 2020

Time: 17:30

Place: Online

Present: CHENG I-Tsun, Prof. Fung

Absent: None

Recorder: CHENG I-Tsun

1. Approval of minutes

This was the first formal group meeting, so there was no minutes to approve.

2. Report on progress

2.1 Researched on potential topics and read research papers on multi-modal tasks (e.g. video retrieval)

3. Discussion items

3.1 Prof. Fung said I can research on Video Retrieval and Video Generation for Fully Automatic Movie Systems since I have worked on part of the project before in the UROP.

3.2 Prof. Fung suggests that I can implement both and compare the two approaches.

4. Goals for the coming week

4.1 Read more on relevant literature about Video retrieval and Video Generation

4.2 Research on previous Automated Movie-related papers

5. Meeting adjournment and next meeting

The meeting was adjourned at 18:00.

6.1.2 Minutes of the 2nd Project Meeting

Date: Sep 14, 2020

Time: 15:30

Place: Online

Present: CHENG I-Tsun, Prof. Chen

Absent: None

Recorder: CHENG I-Tsun

1. Approval of minutes

   The minutes of last meeting were approved without amendment.

2. Report on progress

   2.1 Finished reviewing a lot of research work on Video Retrieval

   2.2 Planning on the proposal report

3. Discussion items

   3.1 Prof. Chen said that the Video Generation technology is still not very developed yet, so it will be really difficult to implement in my project.

   3.2 We reached an agreement that MovieNet is a good dataset to perform video retrieval.

4. Goals for the coming week

   4.1 Finish the proposal report.

   4.2 Begin designing more details for my framework.

5. Meeting adjournment and next meeting

   The meeting was adjourned at 16:00.

6.1.3 Minutes of the 3rd Project Meeting

Date: Jan. 18, 2021

Time: 15:30

Place: Online

Present: CHENG I-Tsun, Prof. Fung

Absent: None

Recorder: CHENG I-Tsun

1. Approval of minutes

   The minutes of last meeting were approved without amendment.

2. Report on progress

   2.1 Finished the Plot Generation Module

   2.2 Working on the Video Retrieval Module

3. Discussion items

   3.1 Prof. Fung said we can try several iterations to match the plot with the videos for Video Retrieval.

   3.2 Prof. Fung also mentioned we can use techniques like abstractive summarization on the combined text of plot and subtitles to form the new plot, and then do retrieval again.

4. Goals for the coming week

   4.1 Try performing more iterations to match and see if the performance is better than one iteration, which is the current approach.

5. Meeting adjournment and next meeting

   The meeting was adjourned at 16:00.

6.1.4 Minutes of the 4th Project Meeting

Date: Feb. 13, 2021

Time: 15:00

Place: Online

Present: CHENG I-Tsun, Prof. Fung

Absent: None

Recorder: CHENG I-Tsun

1. Approval of minutes

   The minutes of last meeting were approved without amendment.

2. Report on progress

   2.1 Finished the iteration approach as proposed by Prof. Fung in the last meeting

   2.2 Finished the entire architecture of our movie generation framework

3. Discussion items

   3.1 Showed Prof. Fung the movies generated using the iteration approach.

   3.2 We think that the iteration approach is better than the non-iteration approach and thus agreed to use this as our final approach for video retrieval.

4. Goals for the coming week

   4.1 Try to come up with an evaluation metric for video retrieval.

5. Meeting adjournment and next meeting

   The meeting was adjourned at 16:30.

6.2 Appendix B: Project Planning

6.2.1 GANTT Chart

Task	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Conduct Literature Survey									
Analyze text generation and video retrieval approaches									
Design Plot Generation Module									
Design Video Retrieval Module									
Design Overall Framework									
Implement Plot Generation Module									
Evaluate Plot Generation Module									
Implement Video Retrieval Module									
Tune the Video Retrieval Algorithm									
Evaluate Video Retrieval Module									
Evaluate Overall Framework									
Write the proposal									
Write the individual essay									
Write the monthly reports									
Write the progress report									
Write the final report									
Prepare for the presentation									
Design the project poster									

6.2.2 Division of Work

CHENG, I-tsun is responsible for everything involved with the project.

6.3 Appendix C: Required Hardware & Software

6.3.1 Hardware

- Development PC: Windows/Mac OSX/Linux
    - Server PC: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
    - Server GPU: GTX 1080Ti 12GPU RAM

6.3.2 Software

- Text editor/IDE (e.g. Visual Studio Code)
- Python 3

6.4 Appendix D: Experiment Details

6.4.1 Bag of Words for Crime, Romance, Sci-Fi

Crime:

crime
criminal
kill
killer
murder
murder
detective
investigate
shoot
dead
plan
case
involved
escape
arrested
officer
office
partner
steal
prison
robbery
fight
gun
jail
drug
scene
body
gangster

Romance:

friendship
beauty
beautiful
pretty
handsome
love
romantic
romance
heart

feelings
propose
fall for
fall in love
kiss
happy
couple
relationship
date
dating
fun
life

Sci-fi:
space
spaceship
spacecraft
aircraft
alien
planet
galaxy
Mars
orbit
Earth
Moon
astronaut
space shuttle
black hole
solar system
satellite
colonize
human
deep space
Operation
species
radiation
oxygen
artificial
artificial intelligence
robot
future
meteor
humanoid
superpower
hero
science
experiment