Automatic script generation could save a considerable amount of resources and offer inspiration to professional scriptwriters. We present **VScript**, a controllable pipeline that generates complete scripts including dialogues and scene descriptions, and presents visually using video retrieval and aurally using text-to-speech for spoken dialogue. With an interactive interface, our system allows users to select genres and input starting words that control the theme and development of the generated script. We adopt a hierarchical structure, which generates the plot, then the script and its audio-visual presentation. We also introduce a novel approach to plot-guided dialogue generation by treating it as an inverse dialogue summarization. Experiment results show that our approach outperforms the baselines on both automatic and human evaluations, especially in terms of genre control.

1 Introduction

As a special literary form used in cinematography and theater, scripts consist of chronological scene descriptions and dialogues of the characters (Kym Spring Linda Burstyn, 2004; Torres and Kallen, 2008; Walker et al., 2012; Owens and Millerson, 2012). Automatic script generation is a challenging task due to three reasons: 1) There are multiple distinct components existing in scripts; 2) The script has to follow a storyline or plot, which serves as a global outline (Fan et al., 2018a; Zhu et al., 2020); 3) The script is required to have rich and diverse content. Building an automated system for scriptwriting would save a substantial amount of human resources and time while bringing inspiration to professional scriptwriters.

Scriptwriter (Zhu et al., 2020) proposes a narrative-guided script generation task, where the provided narrative acts as a global plan to retrieve dialogue utterances. However, this work omits the generation of scene description, which is an important component of a script because they describe the environment and characters. Furthermore, dialogues retrieved from dataset are limited in diversity. Regarding visual presentation, several prior works (Chen et al., 2019; Bensaid et al., 2021) take inspiration from story-boarding and generate a series of images to demonstrate stories. Nevertheless, they do not leverage video, a more informative and attractive medium to present text. In addition, prior works lack the ability to control certain elements, such as genre. Controllable generation systems allow customization of scripts according to user preferences.

In this paper, we present **VScript**, a controllable script generation system. Our work is in contrast to the existing works because 1) our system is able to generate all the essential components of scripts, which include both scene description and dialogues; 2) to better assist the users during the script creation process, **VScript** also offers an audio-visual presentation of the script rather than just images. This presentation can serve as a rough
VScript consists of two modules, i.e., Script Generation and Audio-Visual Presentation. Given the genre and starting words, the Script Generation module generates a genre-specific plot, dialogues, and scene descriptions. The Audio-Visual Presentation module searches for a relevant visualization from a large video database. Audio generated from the dialogue is embedded to give a more immersive presentation.

VScript also incorporates an Audio-Visual Presentation module to vividly demonstrate the script. Using all the information gathered from the Script Generation module, we generate an audio-visual presentation of the script by retrieving video from an automatically constructed video database. Lastly, the generated dialogue is converted into a natural voice by incorporating a text-to-speech (TTS) solution to enrich the user experience.

Our system can serve as a tool for consumers to produce scripts with their preferences. Scriptwriters can also utilize this system to generate scripts and rapidly iterate their ideas. The contributions of this work are threefold:

- To our best knowledge, we are the first to tackle the controllable script generation task, which controls genre and storyline of a script.
- We introduce an effective approach for plot-guided dialogue generation by treating the task as inverse dialogue summarization, which improves the diversity of the generated dialogue while maintaining relevancy to the plot compared to the baselines.
- We explore effective methods to produce an eloquent real-time audio-visual presentation from the generated script.

2 Methodology

As shown in Figure 2, our framework can be decomposed into two modules: script generation and audio-visual presentation. We will describe these two modules in detail in the following sections.

2.1 Script Generation

Instead of directly generating scripts, we first generate a genre-specific plot to guide the dialogue and scene description generation hierarchically. We condition the dialogue on the plot and the scene description on dialogue, instead of the other way around. This is because in film, dialogue between characters changes dynamically to reflect the progression of the plot, while scene descriptions mainly serve to provide extra information telling where and how the dialogue takes place. In this work, we select four classic and popular genres: Crime, Sci-Fi, War, and Romance.
2.1.1 Genre-Specific Plot Generation

As shown in Figure 3, first, we train a class-conditional language model (CC-LM) for plot generation by adopting control codes similar to CTRL (Keskar et al., 2019), where the control codes are a set of predefined prefixes. Then, by training different types of texts with different control codes concatenated in the front, the model is able to learn the correlation between the types and the control codes so that the different control codes could guide the generation process of the language model.

We fine-tune GPT2-large (Radford et al., 2019) on CMU Movie Summary Corpus\(^1\), which contains 42,306 movie plots from Wikipedia and the corresponding metadata, like genre. In our work, the genres of movies are treated as control codes. Each control code guides the generation of episodes of the desired type. The generation probability distribution can be decomposed as:

\[
p(x_t|x_{<t}, c^g) = \prod_{t=1}^{T} p(x_t|x_{<t}, c^g) \tag{1}
\]

\[C = \{c^1, ..., c^g, ..., c^G\}\] denotes the control code, where \(c^g\) means the control code for \(g\)-th genre (\(G\) genres in total).

The CC-LM is trained on a set of plots \(\{x_{1:T}^1, ..., x_{1:T}^n, ..., x_{1:T}^N\}\), where each plot \(x_{1:T}^n\) corresponds with the control code \(c^g \in C\). The training loss is denoted as:

\[
L = - \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(x_t^n|x_{<t}^n, c^g) \tag{2}
\]

**Plot Rescoring** To ensure genre-relevance of the generated plots, we further train a BART-large\(^2\) model as a multi-class genre classifier \(\phi\) on CMU Movie Summary Corpus to verify the genres of the generated plots, so that the classifier can predict the probability of the generated plot belonging to a specific genre \(g\). By leveraging Top-K sampling (Fan et al., 2018a) to generate \(K\) plots \(\{X_1, ..., X_K\}\) with the genre and the starting words, we select the plot \(X_g^*\) with the highest probability over all the generated plots, which is defined as:

\[
X_g^* = \arg \max_{k \in K} p_\phi(y = y_g|X_k) \tag{3}
\]

where \(y = \{y_1, ..., y_g, ..., y_G\}\) denotes the genre classes of the generated plot and \(y_g\) is the class corresponding to genre \(g\).

2.1.2 Plot-Guided Dialogue Generation

The dialogue in the script is required to be casual, natural, and in line with the plot. However, to the best of our knowledge, there is no open-source dataset for plot-to-dialogue generation, and building it would require an intensive human labor. Thus, we treat this task as an inverted abstractive dialogue summarization, where the model is trained to generate the whole dialogue based on the dialogue summary. The model learns to generate the entire dialogue in one fell swoop, which is different from conventional dialogue models generating dialogues turn-by-turn. Two dialogue summarization corpora, SAMSum Corpus (Gliwa et al., 2019) and DialogSum Corpus (Chen et al., 2021b), are combined as the training set. A GPT2-large model is trained on the inverted version of it. During the inference time, we assume that each sentence in the plot can be expanded into a single scene, which can be decomposed into the scene description and the dialogue. We leverage our fine-tuned model to generate dialogues for each plot sentence.

2.1.3 Scene Description Generation

A scene description includes the scene header, i.e., location and time, and the scene context. In order to infer such scene descriptions from each dialogue, we fine-tune the GPT2-large on a paired scene-dialogue corpus. During preprocessing on Film Corpus 2.0\(^3\), we pair up each scene description with its corresponding dialogue to construct the dataset for dialogue-to-scene generation. Finally, we concatenate the scene description and the

\(^1\)http://www.cs.cmu.edu/~ark/personas/
\(^2\)https://huggingface.co/facebook/bart-large-mnli
\(^3\)https://nlds.soe.ucsc.edu/fc2
corresponding dialogues to form a scene. A script is formed by concatenating multiple scenes.

2.2 Audio-Visual Presentation

2.2.1 Video Database Construction

In order to support the need for visual presentation in our system, we construct a video database by crawling news broadcasts and movie recaps from social media. We only preserve the video if i) it has captions, ii) there is a voice-over to introduce and describe what is happening; iii) most of the frames are colorful and include events. The caption of a video is the text describing actions and events in the video and can be used as the label of the video clip to match the plot. Many videos come along with this kind of captions. Motivated by this, captions are also crawled down for matching the video clips and the generated plots.

Post-processing and filtering are further conducted to ensure the quality of the videos. In addition, we classify the video captions by zero-shot text classifier (Yin et al., 2019) to split this database based on genres. The characters and locations in the videos are detected for video retrieval, and the genders of characters are detected to pair different voices generated by TTS. For more details, see the Appendix A. Since our method is zero-shot and independent of the video contents, users can replace the video database with any videos they wish.

2.2.2 Video Retrieval

We use plots instead of scripts to match the video clips because dialogue content, as a major part of a script, could have nothing to do with the video content. For each plot sentence, we retrieve video clips from the video database by calculating the cosine similarity between the sentence embedding of the plot and the corresponding caption with the zero-shot pre-trained DistilRoBERTa-based model from the Sentence Transformers\footnote{https://github.com/UKPLab/sentence-transformers}. We also use the pre-detected gender and location information of the videos to filter out some improper candidates and select the best matching video clip.

2.2.3 Audio Generation

We use a Text-to-Speech API\footnote{https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis} to convert the dialogue text into speech based on the pre-detected gender in pictures and the content of dialogue, which may infer the gender.

3 Experiments

3.1 Baselines

Plot Generation We fine-tune GPT2-large on CMU Movie Summary Corpus and a CC-LM us-
Table 1: Automatic Evaluation for Plot Generation.

| Model                | PPL  | Genre-ACC(%) |
|----------------------|------|--------------|
| GPT2                 | 20.43| -            |
| CC-LM                | 21.98| 63.50        |
| CC-LM+Classifier (Ours) | 21.98| 95.50        |

Plot-Guided Dialogue Generation  We fine-tune DialoGPT-large (Zhang et al., 2020) on the inversed SAMSum and DialogSum Corpus, where the model generates dialogue turn-by-turn iteratively. We fine-tune GPT2-large on the inversed SAMSum and DialogSum Corpus, where the model generates dialogue turn-by-turn (GPT2 T).

Overall Script Generation  In contrast to our proposed pipeline, we fine-tune GPT2-large directly on the scripts from the Film Corpus 1.0 in an end-to-end manner without plot (GPT2 E).

Video Retrieval  We use VideoCLIP (Xu et al., 2021), a pre-trained model for zero-shot video and text understanding to retrieve video based on a plot.

3.2 Evaluation

3.2.1 Automatic Evaluation

Genre-Specific Plot Generation  We score perplexity (PPL) of generated text from our model and baseline (GPT2-large) by another model for fluency evaluation. We use GPT-Neo-1.3B model since it is large for representing the real sentence distribution. We also calculate Genre-ACC, the accuracy of genre control, with the NLI-based zero-shot text classifier. As shown in Table 1, our method can control genre more effectively with only a slight reduction in fluency.

Plot-Guided Dialogue Generation  We evaluate on the test set of SAMSum and DialogSum Corpus. We use BLEU (Papineni et al., 2002) to compare the generated dialogue with the gold-standard human reference. We also calculate Sentence Similarity, which is defined as the cosine similarity between sentence embeddings of plot and dialogue via pre-trained DistilRoBERTa-base model. In addition, we calculate Distinct-n to measure the diversity of generated texts (Li et al., 2016), and Repeat, the average percentage of the uni-grams that occur in the previous 8 tokens (Welleck et al., 2019), to measure the level of repetition. As shown in Table 2, generating the entire dialogue directly rather than turn-by-turn makes the plot-guided dialogues more similar with the gold references and higher semantic similarity with the plot. Both the generated dialogues and the scripts from our model (in Table 3) show higher diversity and lower repetition over the baselines.

3.2.2 Human Evaluation

We conduct human evaluations to further assess the quality of the system using Amazon Mechanical Turk. We randomly select 50 samples per model, each sample is then evaluated by three different annotators to rule out potential bias.

We conduct A/B testing of our framework with the baseline GPT2 E to evaluate generated scripts on Preference and Genre Control. For Human Preference, we ask the annotators to choose which script is the better script from the following aspects: 1) format, whether the text meets the standard of the film script; 2) fluency, whether the writing is smooth and grammatically correct; and 3) consistency, whether the content is logically consistent. For the genre control metric, we ask the annotators to choose which script better belongs to a given genre. In both evaluations, the annotators are given four choices: {neither, both, sample A, or sample B}. As shown in Table 4, human judges prefer the scripts generated by our pipeline. Moreover, our pipeline can control the genre of the script effectively. The human evaluation results are in line with the automatic evaluation results.

For video retrieval, we also conduct A/B testing of our retrieval method with baseline VideoCLIP to evaluate the videos on Relevance. As shown in Table 5, the relevance between the script and video retrieved by our method is slightly higher than that by VideoCLIP. Notably, our method is three orders of magnitude faster than VideoCLIP.
from $\sim$3ms down to $\sim$3$\mu$s per video, which shows better scalability and applicability of our method in a larger-scale system.

### 4 Interactive User Interface

An example of the interaction between users and our system is illustrated in Figure 4. The user interface comprises three main parts: the script area at the top left, the video area at the top right, and the interaction area at the bottom center.

First, users select a genre among Crime, Sci-Fi, War, Romance, and Genre-Free, where Genre-Free means that the model will not control the genre of scripts and the others mean that the model will generate scripts according to the selected genre. Second, users type the starting words into the input box and submit. Finally, the generated script will be displayed in the script area and its audio-visual presentation in the video area. At any time, users can interrupt and choose the genre or input several words again to change the development of the script.

### 5 Related Work

**Story Generation** In recent years, methods for story generation have focused on using deep neural networks and have shown promising results. Skip-Thought Vectors (Kiros et al., 2015) use sentence embeddings to represent stories. DeepTingle (Khalifa et al., 2017) uses recurrent neural networks for story generation. Martin et al. (2018) decompose a story as a sequence of events and apply sequence modelling to generate the story. Fan et al. (2018b) employ a hierarchical approach for story generation by generating a premise and transforming it into a passage. Plan-and-Write (Yao et al., 2019) extracts a storyline composed of keywords and generates a story based on the storyline. Rashkin et al. (2019) generate a narrative using a set of phrases that describe key characters and events in a story.

**Controllable Text Generation Model** In recent years, conditional deep generative models (Kikuchi et al., 2016; Ficler and Goldberg, 2017) have proven to be effective in improving the controllability of the models. CTRL (Keskar et al., 2019) is a class-conditional language model (CC-LM) with 1.6 billion parameters fine-tuned on 50 domains using different control codes as inputs. Plug and Play Language Models (PPLM) Dathathri et al. (2019) combine pre-trained language models and relevant attribute classifiers to steer generation. GeDi (Krause et al., 2020) incorporates a CC-LM as a discriminator to control generation towards the desired attribute.

**Video Retrieval** Video retrieval task has become a long-term research interest (Smeaton et al., 2006), and numerous methods have shown favorable video retrieval quality. Markatopoulou et al. (2017) extract concept words from a video key frame and calculate similarity from the query with the extracted concept words. Li et al. (2019); Bastan et al. (2018); Chen et al. (2018) improve the retrieval quality by extending similarity-based retrieval from a textual space into a learnable latent space. Recent works (Luo et al., 2021; Chen et al., 2021a; Xu et al., 2021) further improve the retrieval quality by utilizing large pre-trained image-text models, such as CLIP (Radford et al., 2021).

### 6 Conclusion

We propose the first controllable script generation framework, **VScript**, that can generate scripts of specific genres and follow the plots. Our framework adopts a hierarchical structure, which generates the plot, then the script and its audio-visual presentation. We adopt inverted abstractive summarization for dialogue generation. According to both automatic and human evaluations, our approach outperforms the baselines, and the effectiveness of genre control is proved. For future work, we would explore fine-grained control of movie script generation, such as specific settings, character personalities, or event details. More exploration on the Video Retrieval or Video Generation can also improve the matching quality between a script and its audio-visual presentation.
7 Ethical Considerations

Since VScript is a framework that covers both linguistic and audio-visual generation, it can bring many possible ethical pitfalls. Throughout the creation of the framework, we tried to mitigate problems that were visible when creating baselines. (Brown et al., 2020; Cahyawijaya et al., 2021) raise awareness about biases that are reflected by large language models. We discover, however, that in the audio-visual generative model the most urgent issue is a removal of sexually explicit, violent contents. Such filtering is necessary to allow larger audiences the access to the generated content.

One of the problems of open-ended script generation is the inability of the model to consistently memorize the intricacies of family relationships and family dependencies. This can be especially problematic in the romance genre, e.g., it can result in the descriptions of close family members falling in love with one another. We create a simple solution, by banning all nouns describing kinship (aunt, father, daughter etc.). In the future, we would like to improve this particular solution to better model complex family relationships without the need of their complete removal. The model is not sensitive towards gender, creating plots, where characters would form a relationship with different genders without consistency, e.g., a male character would marry a female, then a male, then a female again etc., or a male character would marry a female character, and he eventually would get pregnant. To mitigate this problem, we implement a simple function for gender recognition based on the pronouns and gender-specific expressions used, i.e., a pregnant person in our model would be a female. We recognize a simplification of the mode of such gender recognition, as well as an implementation of only two genders. In the future, we plan to work on a better gender recognition model and a more comprehensive gender representation.

To filter any possible curse-words, racial slurs and sexually explicit content, we create lists of banned words that block them from the generated script. A probability of sexually explicit and violent content is especially high because the video frames are retrieved from publicly accessible movies, consequently we filter related key-words from the movie descriptions. In the future, we would like to improve and extend the above-mentioned methods. Furthermore, we believe that a more comprehensive study on bias in script generation is needed for specifying further directions of work on the bias mitigation.

References

Glenn Ballard, Christin Egebjerg, Trond Bolviken, Sigve Endresen, and Brittany Ballard. 2016. Film-making and construction: Two project production systems. In 24th Annual Conference of the International Group for Lean Construction, Boston, Massachusetts, USA.

Muhammet Bastan, Xiangxi Shi, Jiuxiang Gu, Zhao Heng, Chen Zhuo, Dennis Sng, and Alex Chichung Kot. 2018. Ntu rose lab at trecvid 2018: Ad-hoc video search and video to text. In TRECVID.

Eden Bensaid, Mauro Martino, Benjamin Hoover, and Hendrik Strobelt. 2021. Fairytailor: A multi-modal generative framework for storytelling. arXiv preprint arXiv:2108.04324.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aozhu Chen, Fan Hu, Zihan Wang, Fangming Zhou, and Xirong Li. 2021a. What matters for ad-hoc video search? a large-scale evaluation on TRECVID. In 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). IEEE.

Jia Chen, Shizhe Chen, Qin Jin, Alexander G. Hauptmann, Po-Yao Huang, Junwei Liang, Vaibhav, Xiaojun Chang, Jianguo Liu, Ting-Yao Hu, Wenhe Liu, Wei Ke, Wayner Barrios, Haroon Iqrees, Donghyun Yoo, Yaser Sheikh, Ruslan Salakhutdinov, Kris Kitani, and Dong Huang. 2018. Informedia @ TRECVID 2018: Ad-hoc video search, video to text description, activities in extended video. In 2018 TREC Video Retrieval Evaluation, TRECVID 2018, Gaithersburg, MD, USA, November 13-15, 2018. National Institute of Standards and Technology (NIST).
Shizhe Chen, Bei Liu, Jianlong Fu, Ruihua Song, Qin Jin, Pingping Lin, Xiaoyu Qi, Chunting Wang, and Jin Zhou. 2019. Neural storyboard artist: Visualizing stories with coherent image sequences. In Proceedings of the 27th ACM International Conference on Multimedia, pages 2236–2244.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. DialogSum: A real-life scenario dialogue summarization dataset. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5062–5074, Online. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In International Conference on Learning Representations.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018a. Hierarchical neural story generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 899–908, Melbourne, Australia. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In Proceedings of the Workshop on Stylistic Variation, pages 94–104.

Bogdan Gliwa, Iwona Mochol, Maciej Biese, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for ablative summarization. EMNLP-IJCNLP 2019, page 70.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.

Ahmed Khalifa, Gabriella AB Barros, and Julian Togelius. 2017. Deepthing. In 8th International Conference on Computational Creativity, ICCCC 2017, Georgia Institute of Technology.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1328–1338.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shatiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. arXiv preprint arXiv:2009.06367.

Hillary Jordan Kym Spring Linda Burstyn, Pam Cunningham. 2004. Making Movies: A Guide for Young Filmmakers. The Film Foundation.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In HLT-NAACL.

Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. \( w_2 vv + +: \) Fully deep learning for ad-hoc video search. In Proceedings of the 27th ACM International Conference on Multimedia. ACM.

Huashiao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval.

Foteini Markatopoulou, Damianos Galanopoulos, Vasileios Mezaris, and Ioannis Patras. 2017. Query and keyframe representations for ad-hoc video search. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. ACM.

Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shrutna Singh, Brent Harrison, and Mark Riedl. 2018. Event representations for automated story generation with deep neural nets. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Jim Owens and Gerald Millerson. 2012. Video production handbook. Routledge.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5370–5381.
Alan F. Smeaton, Paul Over, and Wessel Kraaij. 2006. Evaluation campaigns and trecvid. In MIR ’06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pages 321–330, New York, NY, USA. ACM Press.

Marco Antonio Torres and Ross Kallen. 2008. A curriculum for digital media creation.

Marilyn A Walker, Grace I Lin, Jennifer Sawyer, et al. 2012. An annotated corpus of film dialogue for learning and characterizing character style. In LREC, pages 1373–1378.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinnan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In International Conference on Learning Representations.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6787–6800.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7378–7385.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3914–3923.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In ACL (demo).

Yutao Zhu, Ruihuha Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. 2020. Scriptwriter: Narrative-guided script generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8647–8657.
A Video Database Construction

A caption is a text describing the action and event in the video and can be used as the label of the video clip to match the plot. The difference between subtitle and caption is that the subtitle containing the dialogue content, which could be nothing to do with the video content. For example, a video shows two women sitting face to face and talking at a café, while the dialogue content is about Star Wars. Thus, we use the plot instead of the script to match the video clips.

Firstly, we use crawl videos and corresponding captions about news broadcasts, and movie recaps from social media using youtube-dl API. Because we match video clip and plot by its caption and title, there are several requirements for the video source: i) The video should have a caption; ii) there is a voice-over to introduce and describe what is happening; iii) and most of the frames are colorful and include events. The frame only consisting of hosts or speakers will be filtered automatically later.

Because some captions are generated automatically word by word, there are no punctuation, and they are not spliced into sentences. We use DeepSegment to sentence-tokenize these captions. To filter the frames including only hosts or speakers and improve the quality of video clips, we use RetinaFace-R50 from InsightFace to detect face. If there is only one face in the centre of the picture which size is within an appropriate range, and it does not move for several frames, we will judge it as a speaker and delete these frames. We also use InsightFace to detect the gender of characters in the videos per second. Furthermore, we use DenseNet-161 from Places365 to recognize the location of scene in the video. Finally, we classify the video captions by NLI-based Zero Shot Text Classifier (Yin et al., 2019) to split this corpus based on genres.

To keep up with time, users can update the Video Dataset to crawl recent videos and popular short videos on their wish.

B Background Music

In order to render atmosphere, we also use different styles and moods of music for different genres of script. For example, the music for crime is rapid and intense, while that for romance is relaxing and soothing.

C Experimental Settings

C.1 Plot Generation

For CC-LM, we fine-tune GPT2-large with control codes (prefixes): “This is a crime/romance/sci-fi/war plot.”. We are using the training hyperparameters: the learning rate is 3e-5, AdamW optimizer, and WarmupDecayLR scheduler and generate plot using top-k (k=4) sampling. For Genre-Classifier, we fine-tune BART-large with the same training hyperparameters: the learning rate is 3e-5, AdamW optimizer, and WarmupDecayLR scheduler. For GPT2 baseline, we fine-tune the model with the same hyperparameter setting as GPT2-large models in our pipeline.

C.2 Plot-Guided Dialogue Generation

The model in this stage of our pipeline has the same training and generation hyperparameters as GPT2-large model in Sec C.1. For DialogGPT baseline, we fine-tune the model with learning rate (5e-5) and the other hyperparameters are the same as GPT2-large models in our pipeline. For GPT2 Turn-by-Turn (GPT2 T) baseline, we use the same hyperparameter setting as GPT2-large models in our pipeline.

C.3 Scene Description Generation

The model in this stage of our pipeline has the same hyperparameters as GPT2-large model in Sec C.1.

C.4 Overall Script Generation

For GPT2_E baseline, we use the same hyperparameter set with GPT2-large model in Sec C.1.