VScript: Controllable Script Generation with Visual Presentation

Ziwei Ji, Yan Xu, I-Tsun Cheng, Samuel Cahyawijaya, Rita Frieske,
Etsuko Ishii, Min Zeng, Andrea Madotto, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Hong Kong University of Science and Technology
zjiad@connect.ust.hk, pascale@ece.ust.hk

Abstract

In order to offer a customized script tool and inspire professional scriptwriters, we present **VScript**. It is a controllable pipeline that generates complete scripts, including dialogues and scene descriptions, as well as presents visually using video retrieval. 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 first generates the plot, then the script and its visual presentation. A novel approach is also introduced to plot-guided dialogue generation by treating it as an inverse dialogue summarization. The experiment results show that our approach outperforms the baselines on both automatic and human evaluations, especially in genre control.

1 Introduction

Artificial intelligence (AI) introduces significant changes in the creation of artworks, such as stylistic painting (Kotovenko et al., 2019), poem writing (Hu and Sun, 2020), music composition (Dong et al., 2018), and converting the perception of creativity. In particular, AI can assist in streamlining the art-making process and give humans fresh inspirations (Anantrasirichai and Bull, 2021), and one plausible application is scriptwriting. As a specific literary form, the script is indispensable in cinematography and theater (Owens and Millerson, 2012; Walker et al., 2012). To enable the collaboration between scriptwriter and AI, an automatic script generation system must equip with three aspects. First, the system must generate a complete script consisting of chronological scene descriptions and dialogues. Second, the system should provide controllability, e.g., customize script genre or storyline (Cavazza and Young, 2017). Third, as a creative work, the generated script is required to have rich and diverse content.

While Zhu et al. (2020) propose a narrative-guided script generation task, they only focus on retrieving dialogue utterances and omit scene description, an essential component of a script describing the environment and characters. Other works (Chen et al., 2019; Bensaid et al., 2021) take inspiration from storyboarding and utilize a series of images to demonstrate stories. Nevertheless, no prior work presents scripts by leveraging video, a more informative and attractive medium. In addition, prior works lack the ability to control certain elements, such as genre. Controllable generation systems (Keskar et al., 2019; Dathathri et al., 2019; Madotto et al., 2020) can help to allow the customization of scripts based on user preferences.

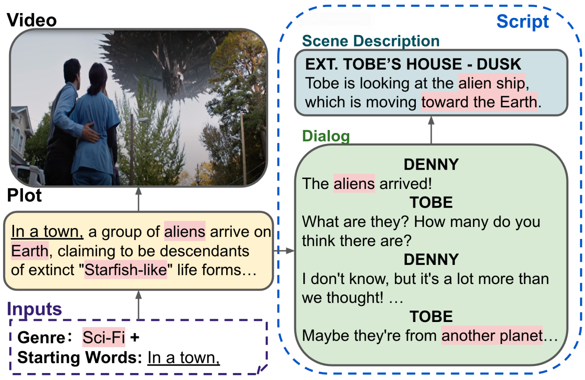

Video
Plot
In a town, a group of aliens arrive on Earth, claiming to be descendants of extinct "Starfish-like" life forms…
Inputs
Genre: Sci-Fi +
Starting Words: In a town,
Script
Scene Description
EXT. TOBE'S HOUSE - DUSK
Tobe is looking at the alien ship, which is moving toward the Earth.
Dialog
DENNY
The aliens arrived!
TOBE
What are they? How many do you think there are?
DENNY
I don't know, but it's a lot more than we thought! …
TOBE
Maybe they're from another planet…

Figure 1: An example of the generated script (right) with its visual presentation (top left) from **VScript**. Given the inputs, i.e., genre and starting words, a plot is generated, which guides the generation of a script consisting of a scene description and a dialogue. The words highlighted in pink show the belongingness to the given genre (Sci-Fi). Additionally, the video vividly presents the script.

In this paper, we present **VScript**, a controllable script generation system that includes all essential components of a script. We adopt a hierarchical structure for our framework by implementing high-level planning (Fan et al., 2018) and following the guidelines of video-making processes (Owens and Millerson, 2012). As shown in Figure 2, **VScript** is composed by **Script Generation** and **Visual Presentation** modules. The examples of correspond-

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations, pages 1–8
November 20–23, 2022. ©2022 Association for Computational Linguistics

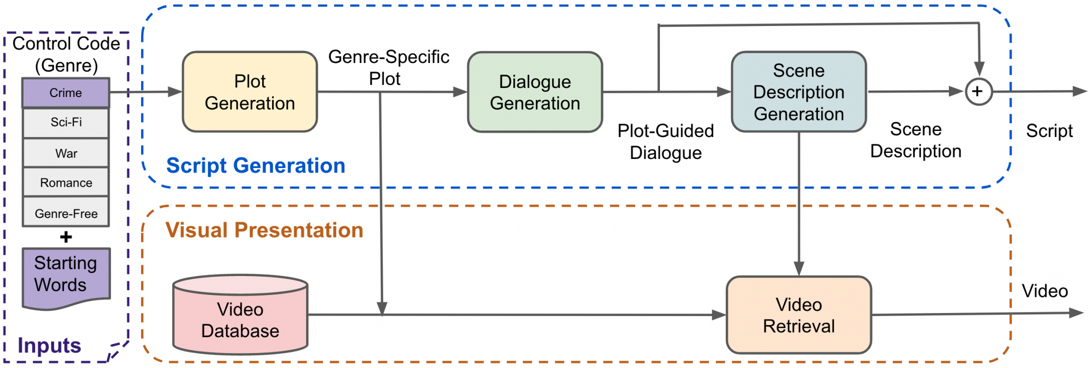


Figure 2: **VScript** consists of two modules, i.e., **Script Generation** and **Visual Presentation**. Given the genre and starting words, the Script Generation module generates a genre-specific plot, dialogues, and scene descriptions. The Visual Presentation module searches for a relevant visualization from a large video database.

ing components are in Figure 1. **Script Generation** module consists of three sub-modules: 1) plot generation, 2) dialogue generation, and 3) scene description generation. Firstly, the system generates a genre-specific plot as an outline of the script. To avoid generating aimless scripts, the system allows users to provide some brief initial information, i.e., genre and the beginning of the plot, for a high degree of control over the script on the genre and story trend. Secondly, we conduct a zero-shot plot-guided dialogue generation by framing the problem as an inverse task of abstractive dialogue summarization. Finally, we generate scene descriptions based on the dialogues to produce complete scripts. **Visual Presentation** module vividly demonstrates the script by retrieving videos from an automatically constructed video database. It can serve as a rough visual draft to improve users' engagement and help scriptwriters rapidly iterate ideas.

To our best knowledge, we are the first to tackle the controllable script generation task, which controls a script's genre and storyline. We also introduce a practical 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. In addition, we explore effective methods to produce an eloquent real-time visual presentation from the generated script. According to the evaluation results, our scripts are controllable and preferred by humans. Thus, **VScript** can serve as a tool for users to produce scripts with their preferences and for scriptwriters to optimize the writing process. We think **VScript** has the potential to promote AI-human collaboration in script generation. Limitations and Ethical Considerations are discussed in Appendix A and B.

2 Related Work

Story Generation In recent years, methods for story generation have focused on using neural networks and have shown promising results. Martin et al. (2018) decompose a story as a sequence of events and apply sequence modelling to generate the story. Fan et al. (2018) employ a hierarchical 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. Lovenia et al. (2022) generate a story using a genre and an image as its context.

Controllable Text Generation Model Conditional deep generative models are effective in improving the controllability of the models. CTRL (Keskar et al., 2019) is a class-conditional language model (CC-LM) pre-trained on 50 domains with different control codes. Plug and Play Language Models (PPLM) Dathathri et al. (2019) combine pre-trained language models and attribute classifiers to steer generation. GeDi (Krause et al., 2020) incorporates CC-LM as a discriminator to control generation towards the desired attribute.

3 Methodology

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

3.1 Script Generation

A plot, or so-called outline, is a vital component required for professional script writing, which specifies a series of headings showing the main themes that need to be discussed (Owens and Millerson, 2012). Following this concept, we first generate a genre-specific plot to hierarchically guide the dialogue and scene description generation. We condition the dialogue on the plot and the scene description on dialogue instead of the other way around since dialogues between characters change dynamically to reflect the progression of the plot, while scene descriptions mainly provide detailed information about where and how the dialogue takes place. In this work, we select four classic and popular genres for the plot generation, i.e., `Crime`, `Sci-Fi`, `War`, and `Romance`.

3.1.1 Genre-Specific Plot Generation

Inspired by CTRL (Keskar et al., 2019), we first train a class-conditional language model (CC-LM) for plot generation by adopting control codes which are a set of predefined genres. Then, by training different types of text with different control codes concatenated in the front, the model can learn the correlation between the types and the control codes so that the different control codes can 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, such as 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. See Appendix C for more details.

Plot Rescoring To ensure the consistency of the generated plots and the target genre, we further train a multi-class genre classifier that can predict the probabilities of a plot belonging to each genre. Then, we generate N plots with the same genre and starting words via Top-K sampling. Finally, we select the plot with the highest probability among all the generated N plots.

3.1.2 Plot-Guided Dialogue Generation

The dialogue in the script is required to be casual, natural, and in line with the plot. However, to our knowledge, there is no open-source dataset for plot-to-dialogue generation, and building it would require intensive human labour. Thus, we treat this task as an inversed 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 (Gliwa et al., 2019) and DialogSum Corpus (Chen et al., 2021), are combined as the training set. A GPT2-large model is trained on the inversed 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.

3.1.3 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[2], we pair each scene description with its corresponding dialogue to construct the dataset for dialogue-to-scene generation. Finally, we concatenate the scene description and the corresponding dialogues to form a scene. A script is formed by concatenating multiple scenes.

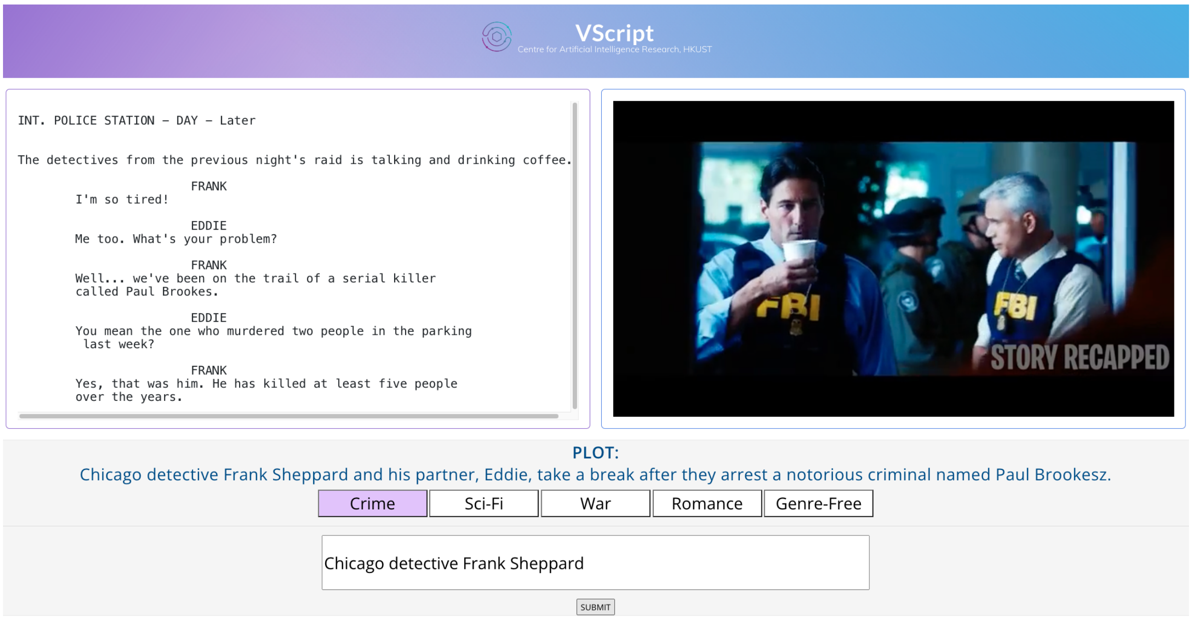
3.2 Visual Presentation

In addition to the generated script, we provide a visual presentation, which can serve as a rough visual draft for the users. We retrieve a video clip whose caption describes similar actions or events to the generated script. Note that we only utilize the visual contents of the retrieved video and ignore the auditory information. This disregard is intended to enhance retrieval quality, as the conversational content and visual appearance of a video are often inconsistent. For example, a video shows two women sitting face to face at a café, looking bright and peaceful, while their conversation is about fierce interstellar wars. In this section, we explain our video database construction process and the retrieval method.

Video Database Construction We construct a video database including news broadcasts, documentaries, and movie recaps from social media and only preserve the video if 1) it has captions; 2) there is a voice-over to introduce and describe what is happening; 3) most of the frames have rich content. Post-processing and filtering are further conducted

[1] `http://www.cs.cmu.edu/~ark/personas/`

[2] `https://nlds.soe.ucsc.edu/fc2`



```
INT. POLICE STATION - DAY - Later

The detectives from the previous night's raid is talking and drinking coffee.

                        FRANK
        I'm so tired!

                        EDDIE
        Me too. What's your problem?

                        FRANK
        Well... we've been on the trail of a serial killer
        called Paul Brookes.

                        EDDIE
        You mean the one who murdered two people in the parking
         last week?

                        FRANK
        Yes, that was him. He has killed at least five people
        over the years.
```

PLOT:
Chicago detective Frank Sheppard and his partner, Eddie, take a break after they arrest a notorious criminal named Paul Brookesz.

Chicago detective Frank Sheppard

Figure 3: The **VScript**'s user interface. There are three main areas: script area (top left), video area (top right), and interaction area (bottom centre). For example, a user chooses "`Crime`", and input "`Chicago detective Frank Sheppard`". Then, the generated script and its visual presentation are displayed separately.

to ensure the quality of the videos. The characters (including number and gender), time (day/night) and locations in the videos are detected for video retrieval. In addition, we classify the video captions by a zero-shot text classifier [3] to split this database based on genres. For more details, see Appendix D. Since our method is zero-shot and independent of the video contents, users can replace the video database with any preferred videos.

Video Retrieval To match the video clips with the generated scripts, we use plots as queries for video retrieval since there may be trivial and lengthy dialogues in scripts that affect the coherence among the retrieved video clips. 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 pre-trained DistilRoBERTa-based model [4]. We also use the videos' pre-detected gender and location information to filter out some improper candidates and select the best matching video clip.

4 Interactive User Interface

An example of the interaction between users and **VScript** is illustrated in Figure 3. The user interface comprises three parts: the script area (top left), the video area (top right), and the interaction area (bottom centre). First, users select the script genre among `Crime`, `Sci-Fi`, `War`, `Romance`, and `Genre-Free`. 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 visual presentation in the video area. Users can interrupt at any time and choose the genre or input some words to steer the script's development.

5 Experiments

5.1 Baselines

Plot Generation We fine-tune **GPT2**-large on CMU Movie Summary Corpus and a **CC-LM** using GPT2-large backbone on the same corpus without the genre-classifier.

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[5] in an end-to-end manner without plot **(GPT2_E)**.

[3] https://huggingface.co/facebook/bart-large-mnli

[4] https://github.com/UKPLab/sentence-transformers

[5] https://nlds.soe.ucsc.edu/fc

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

Table 1: Automatic Evaluation for Plot Generation.

Model	BLEU	Sent Sim	Repeat (%)	Dist-n (n=1,2,3)
DialoGPT	13.35	54.18	20.25	1.7/13.6/42.69
GPT2 T	13.37	55.34	18.73	1.73/14.29/43.89
GPT2 (Ours)	16.4	58.97	9.68	3.19/22.4/53.32

Table 2: Automatic Evaluation for Plot-Guided Dialogue Generation.

Model	Dist-n (n=1,2,3)	Repeat (%)
GPT2_E	8.42/36.19/68.46	12.52
Ours	5.47/35.81/73.3	4.35

Table 3: Automatic Evaluation for Scripts.

Video Retrieval To verify our video retrieval, we use **VideoCLIP** (Xu et al., 2021), a pre-trained model for zero-shot video and text understanding.

5.2 Evaluation

5.2.1 Automatic Evaluation

Genre-Specific Plot Generation We score perplexity (**PPL**) of texts generated from our model and baseline (GPT2-large) by another model for fluency evaluation. We use the GPT-Neo-1.3B model since it is large enough to represent 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 models on the test set of SAMSum and DialogSum. We use **BLEU** 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 [6] of plot and dialogue. In addition, we calculate **Distinct-n** to measure the diversity of generated texts, and **Repeat**, the average percentage of the unigrams 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 to 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.

Ours vs GPT2-E	Win(%)	Loss(%)	Tie(%)
Preference	54.00	27.33	18.67
Genre Control	95.33	4.00	0.67

Table 4: Human Evaluation for Scripts.

Ours vs VideoCLIP	Win(%)	Loss(%)	Tie(%)
Relevance	27.33	13.33	59.33

Table 5: Human Evaluation for Video Retrieval.

5.2.2 Human Evaluation

We conduct human evaluations further to assess the quality of **VScript** using Amazon Mechanical Turk. We randomly select 50 samples per model, and three annotators then evaluate each sample to rule out potential bias. We conduct A/B testing against the baseline GPT2_E to assess generated scripts on **Preference** and **Genre Control**. For **Preference**, we ask the annotators to choose which script is the better one from three aspects [7]: 1) format, whether the text meets the standard of film scripts; 2) fluency, whether the writing is smooth and grammatically correct; and 3) consistency, whether the content is logically consistent. For **Genre Control**, we ask the annotators to choose which script better belongs to a given genre. In both tests, the annotators are given four choices: {neither, both, sample A, or sample B}. As shown in Table 4 [8], human judges prefer the scripts generated by **VScript**, which is inline with the automatic evaluation. For video retrieval, we conduct A/B testing against VideoCLIP to evaluate the **Relevance**. The **Relevance** between the script and video retrieved by **VScript** is slightly higher than the baseline, as in Table 5 [8].

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 visual presentation. We adopt inversed abstractive summarization for dialogue generation. Based on our experiments, **VScript** outperforms the baselines, and its effectiveness in genre control is proven.

[6] https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v2

[7] Please refer to Appendix G for the results of each aspect.

[8] The result is statistically significant with $p < 0.05$.

References

Nantheera Anantrasirichai and David Bull. 2021. Artificial intelligence in the creative industries: a review. *Artificial Intelligence Review*, pages 1–68.

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

Marc Cavazza and R Michael Young. 2017. Introduction to interactive storytelling. *Handbook of digital games and entertainment technologies*.

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. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, et al. 2019. Plug and play language models: A simple approach to controlled text generation. In *ICLR*.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.

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

Jinyi Hu and Maosong Sun. 2020. Generating major types of chinese classical poetry in a uniformed framework. In *Proceedings of the 12th Language Resources and Evaluation Conference*.

Oana Ignat, Santiago Castro, Hanwen Miao, Weiji Li, and Rada Mihalcea. 2021. Whyact: Identifying action reasons in lifestyle vlogs. In *EMNLP-IJCNLP*.

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*.

Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Bjorn Ommer. 2019. A content transformation block for image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10032–10041.

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

Holy Lovenia, Bryan Wilie, Romain Barraud, Samuel Cahyawijaya, Willy Chung, and Pascale Fung. 2022. Every picture tells a story: Image-grounded controllable stylistic story generation. *arXiv preprint arXiv:2209.01638*.

Andrea Madotto, Etsuko Ishii, Zhaojiang Lin, Sumanth Dathathri, and Pascale Fung. 2020. Plug-and-play conversational models. In *Findings of EMNLP 2020*.

Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. 2018. Event representations for automated story generation with deep neural nets. In *AAAI*.

Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. Probing toxic content in large pre-trained language models. In *ACL-IJCNLP 2021*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *ACL*.

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*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, et al. 2019. Neural text generation with unlikelihood training. In *ICLR*.

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 *EMNLP-IJCNLP*.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *AAAI*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *EMNLP-IJCNLP*.

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, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. 2020. Scriptwriter: Narrative-guided script generation. In *ACL 2020*, pages 8647–8657.

A Limitations

The performance of video retrieval is limited to some extent by the quality of the sentence embedding, face detection, and place detection via off-the-shelf tools. The effect of visual presentation depends heavily on the quality and quantity of the video database. More exploration in video retrieval or video generation can also improve the matching quality between a script and its visual presentation. In addition, we would explore a more fine-grained control, such as specific settings, character personalities, or event details for future work.

B Ethical Considerations

Copyright We collect publicly available YouTube videos using the official YouTube API and follow the typical processing procedures (Ignat et al., 2021). We use the muted video footage and are neutral to the opinions expressed therein. We will not release the database and are accountable for violating other parties' rights or terms of service.

Toxic Content **VScript** leverages large language models, which raise awareness of carrying biases and toxic content (Ousidhoum et al., 2021). Therefore, we create lists of banned words that block them from the generated script to filter the possible curse words, racial slurs and sexually explicit contents. Since our videos are retrieved from publicly accessible media, which could include bloody and erotic content, we also filter these video clips based on the descriptions and captions.

C Genre-Specific Plot Generation

As shown in Figure 4, 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 follows:

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

$\mathbf{C} = \{c^1, ..., c^g, ..., c^G\}$ 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 $\{x^1_{1:T^1}, ..., x^n_{1:T^n}, ..., x^N_{1:T^N}\}$, where each plot $x^n_{1:T^n}$ corresponds with the control code $c^x \in \mathbf{C}$. The training loss is denoted as:

$$L = -\sum_{n=1}^{N}\sum_{t=1}^{T^n} \log p_\theta(x^n_t|x^n_{<t}, c^x) \tag{2}$$

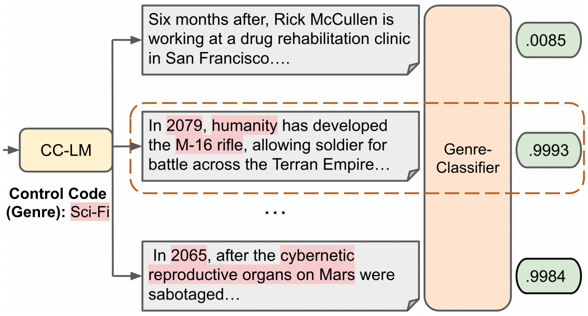

CC-LM
Control Code (Genre): Sci-Fi
Six months after, Rick McCullen is working at a drug rehabilitation clinic in San Francisco....
In 2079, humanity has developed the M-16 rifle, allowing soldier for battle across the Terran Empire...
...
In 2065, after the cybernetic reproductive organs on Mars were sabotaged...
Genre-Classifier
.0085
.9993
.9984

Figure 4: The process of plot generation. The CC-LM generates several candidates, and then the Genre-Classifier scores and ranks them to select the one that most belongs to the expected genre.

Plot Rescoring To verify the genres of the generated plots, we train a multi-class genre classifier ϕ to predict the probability of the generated plot belonging to a specific genre g. By leveraging Top-K sampling, we generate $N(N = 10)$ plots $\{X_1, ..., X_n, ...X_N\}$ with the genre and the starting words. Finally, we select the plot X^*_g with the highest probability over all the generated plots, which is defined as:

$$X^*_g = \arg\max_{n \in \mathbf{N}} p_\phi(\mathbf{y} = y_g|X_n) \tag{3}$$

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

D Video Database Construction

Firstly, we obtain videos and corresponding captions about news broadcasts and movie recaps from YouTube via the official API. Since some captions are generated automatically word by word, there is no punctuation, and they are not spliced into sentences. We use DeepSegment[9] to sentence-tokenize these captions. To ensure the richness of the video image content and improve the quality of video clips, the frames only consisting of hosts or speakers are filtered automatically. We use RetinaFace-R50 from InsightFace [10] to detect face. If there is only one face in the centre of the picture whose size is within an appropriate range, and it does not move

[9] https://github.com/notAI-tech/deepsegment

[10] https://github.com/deepinsight/insightface

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. We use DenseNet-161 from Places365 [11] to recognize the location of scene in the video. Furthermore, in order to match the time in scene description, we train a Vision Transformer (ViT) [12]-based day/night image classifier on Aachen Day-Night Dataset [13], AMOS Day-Night Dataset [14], and [15]. Finally, we classify the video captions by NLI-based Zero Shot Text Classifier (Yin et al., 2019) to split this corpus based on genres. Since our method is zero-shot and independent of the video contents, users can update or replace the video database on their wish.

E Background Music

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

F Experimental Settings

F.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 use the training hyperparameters: the learning rate is 3e-5, AdamW optimizer, and WarmupDecayLR scheduler and generate plots 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 the GPT2 baseline, we fine-tune the model with the same hyperparameter setting as GPT2-large models in our pipeline.

F.2 Plot-Guided Dialogue Generation

The model in this stage of our pipeline has the same training and generation hyperparameters as the GPT2-large model in Appendix F.1. For the DialogGPT baseline, we fine-tune the model with a 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 the GPT2-large models in our pipeline.

Model	Format (%)	Fluency	Consistency
GPT2-E	93(0.26)	2.92(0.45)	2.85(0.45)
VScript	**95(0.22)**	**3.06(0.42)**	**3.00(0.50)**

Table 6: Additional human evaluation for scripts.

F.3 Scene Description Generation

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

F.4 Overall Script Generation

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

G Additional Human Evaluation

In addition, we also evaluate the quality of each generated script for a more comprehensive study, compared with GPT2-E model. As mentioned in Section 5.2.2, we breakdown the **Preference** metric into three aspects: **Format**, **Fluency**, and **Consistency**. **Format** measures whether our generation meets the standard of the script, which is defined as a text that contains a scene header, and dialogue (including monologue). For this metric, we conduct True/False binary evaluation. **Fluency** reflects whether the writing is smooth and non-repetitive, without grammatical and spelling mistakes. **Consistency** emphasizes whether the content is logically consistent. For Fluency and Consistency, we leverage a 4-point Likert Scale, where 1 indicates non-fluent/inconsistent and 4 indicates a very fluent/consistent text. For each model, we randomly sample 50 generated scripts from each model. And each script is evaluated by three annotators. An individual t-test is conducted for significance validation of the human evaluation results. As shown in Table 6 [16], while achieving script formulation, our system has slightly higher fluency than the baseline model, indicating that fluency is not compromised. Our framework is also more consistent and logical, with considerably higher consistency than GPT2-E.

[11] https://github.com/CSAILVision/places365

[12] https://huggingface.co/google/vit-base-patch16-224-in21k

[13] https://www.visuallocalization.net/

[14] https://www.kaggle.com/datasets/stevemark/daynight-dataset

[15] https://github.com/kushagra2jindal/DayNightClassificationModel

[16] The result is statistically significant with $p < 0.05$.