GPT-2-based Human-in-the-loop Theatre Play Script Generation[*]

Rudolf Rosa,[μ] **Patrícia Schmidtová,**[μ] **Ondřej Dušek,**[μ] **Tomáš Musil,**[μ] **David Mareček,**[μ]
Saad Obaid,[μ] **Marie Nováková,**[σμ] **Klára Vosecká**[δ] and **Josef Doležal**[δ]
[μ]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[σ]The Švanda Theatre in Smíchov, Prague, Czechia
[δ]The Academy of Performing Arts in Prague, Theatre Faculty (DAMU), Prague, Czechia
rosa@ufal.mff.cuni.cz

Abstract

We experiment with adapting generative language models for the generation of long coherent narratives in the form of theatre plays. Since fully automatic generation of whole plays is not currently feasible, we created an interactive tool that allows a human user to steer the generation somewhat while minimizing intervention. We pursue two approaches to long-text generation: a flat generation with summarization of context, and a hierarchical text-to-text two-stage approach, where a synopsis is generated first and then used to condition generation of the final script. Our preliminary results and discussions with theatre professionals show improvements over vanilla language model generation, but also identify important limitations of our approach.

1 Introduction

Natural language generation (NLG) is currently dominated by large pre-trained language models, such as GPT-3 (Brown et al., 2020). The models show especially strong performance in generating short to medium length in-domain texts, such as news stories, which fit into the window size of the model (e.g. 512 or 1,024 subword tokens). Successfully handling significantly larger and/or out-of-domain documents is a matter of ongoing research (Beltagy et al., 2020; Zaheer et al., 2020; Gururangan et al., 2020; Chen et al., 2020).

In the THEaiTRE project, we focus on generating theatre play scripts. This task combines the challenges of narrative generation (Riedl, 2016) and dialogue generation (Wen et al., 2016), and could be seen either as generating dialogues with a very large context, or as generating a narrative in the form of a dialogue. Additional challenges include the complex structure of the theatre scripts (including setting descriptions, dialogue lines with character names, and stage directions), their very large length, their pseudo-multi-author nature (as lines pertaining to different characters use different styles and represent different standpoints), or the low availability of large in-domain datasets.

We investigate the capabilities of current NLG approaches on this task. Specifically, we use and adapt current large pre-trained neural language models and employ other relevant natural language processing (NLP) techniques to adjust the existing approaches and tools to the theatrical script domain.

Our aim is to produce a mostly automatically generated play, with minimal human-in-the-loop interventions, and have the generated play rehearsed and staged by a theatre. We build upon our previous work (Rosa et al., 2021), where we produced a generated play by using vanilla GPT-2 and generated individual, loosely connected scenes, but now aim at full play generation. In order to do so, we explore a two-phase hierarchical text-to-text approach, where a synopsis is generated first and then used as a basis for subsequent generation of scenes. We compare this method to a flat generation approach with summarization, which is similar to our previous work (Rosa et al., 2021). We use models finetuned on in-domain theatre or movie scripts to better fit the domain, and we allow minimal but precise human intervention using a custom-built web-based interface: regenerating a line, choosing the next character to speak, deleting or inserting a generated or a human-written line into the script. All human interventions are recorded. A simplified demo version of the tool used for the generation is freely available online.[1] We include preliminary intrinsic evaluation and discuss qualitative feedback given by the theatre professionals. Our results support finetuning and more precise human intervention; however, the two-stage hierarchical approach shows difficulties following the pre-generated synopsis.

[*] Archival WNU submission.

[1] https://theaitre.com/demo

Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022), pages 29 - 37
July 15, 2022 ©2022 Association for Computational Linguistics

2 Related Work

Our approach is inspired by the work of Fan et al. (2018) and Fan et al. (2019), who propose a hierarchical system for story generation. A similar idea has been explored by Rashkin et al. (2020), who generate a story conditioned on a given outline. Tan et al. (2021) approach long text generation by generating domain-specific words first and then iteratively refining it until whole sentences are formed. Unlike these works, we generate scripts rather than stories, i.e. not prose but dialogues, which are also longer than typical stories. For dialogue generation, Xu et al. (2021)'s work is close to our baseline flat approach (Section 3) in that they generate long dialogues by using summarization.

A few works also investigate human-machine interaction during text generation, with different aims from ours: Roemmele (2021) investigates how automatically generated texts can inspire human writing. Akoury et al. (2020) use the amount of required human post-editing as a story quality metric.

A number of language generation tools is available online, both free and paid, typically based on GPT-2 and GPT-3 language models (Radford et al., 2019; Brown et al., 2020), sometimes trained or fine-tuned for a specific domain or task. Prominent examples include news generators such as *Grover*[2] by Zellers et al. (2019) or *News You Can't Use*[3] by Geitgey (2019), the text adventure game *AI Dungeon*,[4] the code completion tools *GitHub Copilot*[5] or *Deep Tabnine*,[6] and chatbots such as *AI|Writer* or *Project December*.[7] However, to the best of our knowledge, no generation tool has been released specifically for theatre scripts.

There have been several other projects using automatically generated scripts, including *Beyond the Fence*, a musical based on suggestions from several automated tools (Colton et al., 2016), *Sunspring*, a short sci-fi movie with an LSTM-generated script (Benjamin et al., 2016), *Lifestyle of the Richard and Family*, a theatre play written with the help of a next word suggestion tool (Helper, 2018), or the performances of the *Improbotics* group who improvise on stage with real-time GPT-3-generated lines (Mathewson and Mirowski, 2017). However, the tools used in these projects are not publicly available online, and often there is little transparency about the particularities of the exact design and usage of the tools. Moreover, these projects typically use substantial human curation.

Domain	# Scripts	Avg. # Lines	Avg. # Sentences
Movies	1,067	783	2,537
TV Shows	6,057	314	902
Theatre	5,517	530	1,529
All	12,641	446	1,310

Table 1: A brief overview of the script dataset we use for finetuning.

3 Flat Generation with Summarization

The flat generation variant is based on our previous approach (Rosa et al., 2021) of using a standard generative model but employing extractive text summarization to deal with the limited window (1,024 tokens for GPT-2) so that longer scripts can be generated without the loss of the global context. Instead of using a vanilla GPT-2 model as in our previous work, we finetune our models on a large collection of ca. 12k theatre and movie scripts. The domains and volumes of data can be found in Table 1. The theatre plays and TV shows scripts were scraped from various websites, the movie section comes from (Lison and Meena, 2016).

The operation of flat generation looks as follows: the user inputs a scene setting, character names and their first lines, from which we construct the input prompt in the following format:

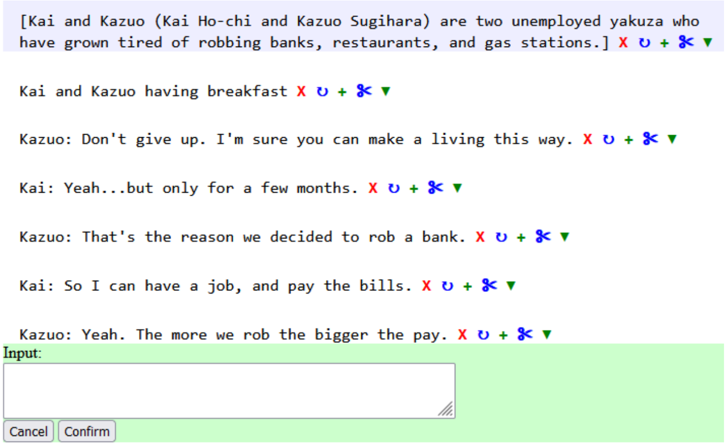
```
Scene setting.

Character Name: Character line.

Character Name: Character line.
```

The model then generates a continuation of the script line by line (see Figure 2).[8] At each step, the user can choose whether they want to regenerate the last generated line (i.e. generate a different continuation), or whether they want to continue by generating a further line. They can also choose the next character and let the model generate their line, or insert/delete lines within the generated text. A screenshot of this tool is presented in Figure 1.

[2] https://rowanzellers.com/grover/
[3] https://newsyoucantuse.com/
[4] https://play.aidungeon.io/
[5] https://copilot.github.com/
[6] https://www.tabnine.com/
[7] https://projectdecember.net/



[Kai and Kazuo (Kai Ho-chi and Kazuo Sugihara) are two unemployed yakuza who have grown tired of robbing banks, restaurants, and gas stations.] X ↻ + ✂ ▼

Kai and Kazuo having breakfast X ↻ + ✂ ▼

Kazuo: Don't give up. I'm sure you can make a living this way. X ↻ + ✂ ▼

Kai: Yeah...but only for a few months. X ↻ + ✂ ▼

Kazuo: That's the reason we decided to rob a bank. X ↻ + ✂ ▼

Kai: So I can have a job, and pay the bills. X ↻ + ✂ ▼

Kazuo: Yeah. The more we rob the bigger the pay. X ↻ + ✂ ▼

Figure 1: A screenshot of the tool used for the generation. The cross generates an alternative continuation starting with the given line. The arrow creates an alternative line while keeping the script continuation. The plus symbol generates and inserts a line, while the scissors symbol deletes it without any changes to the continuation in both cases. Finally, the triangle symbol allows for human input that prompts the regeneration of the continuation.

Polonius speaks to the king. Enter Hamlet.	
Polonius:	I hear him coming; let's hide, sir.
Hamlet:	To be or not to be; that is the question.
Polonius:	That is the reason, Hamlet. You are a good man.
Hamlet:	I am but what I am.
Polonius:	You are a bad man.

Figure 2: Flat script generation example. The initial human-written prompt is shown above the dividing line, the following generated outputs follow below.

4 Two-phase Hierarchical Generation

Our second, newly developed approach is a two-phase text-to-text hierarchical generation approach: (1) a synopsis is generated from a user-supplied play title, (2) the play dialogue is generated, conditioned on the title and a part of the synopsis. Both phases use specific models finetuned on our in-domain datasets.

The input for the first phase is the title of the play, from which the synopsis of the play is generated (Figure 3, Section 4.1). At each step, The user has the option of continuing generation, regenerating or deleting lines (roughly corresponding to sentences). Once the user is satisfied with the generated synopsis (or the generation ends by the model generating the *endoftext* token), the synopsis is used as input for the second phase.

In the second phase, shown in Figure 4, the play script is generated from the synopsis (see Section 4.2). The user is now provided with a set of options similar to the flat approach: at each step choosing between generating a character line (and potentially also choosing which character should speak the line) or moving on to the next part of the generated synopsis.

A notebook full of ideas was stolen from an unbuttoned backpack by a mailman in a blue vest.

A man named Tom, a man in a trench coat, arrives to deliver the notebook.
Tom tries to leave, but is interrupted by the arrival of his mailman brother, Jerry, in his mail truck.
Jerry offers to take Tom to his house, but Tom is determined to deliver the notebook. As Tom drives through the cornfield, he accidentally stops at a house, which he mistakenly assumes is inhabited by the owner, a widowed woman named Marjorie.
She tells him she is waiting for Tom to come home, and she and Tom go into the house together.
Jerry arrives and finds Tom's truck with the notebook, having accidentally left it in the truck while searching for Jerry, and is surprised and angry to find Marjorie there.

Figure 3: An example for hierarchical generation 1st phase: title to synopsis (input title shown above the dividing line, the play follows below).

4.1 1st Phase: Synopsis generation

The goal of this phase is to generate a synopsis based on a user-specified play title. For this, we finetune pretrained language models on a dataset consisting of synopses of theatre plays (scraped by us from Wikipedia), movies (Robischon, 2018; Kar et al., 2018), TV series (scraped by us from various fan wiki pages) and books (Bamman and

[8] The GPT-2 model sticks to the input format and generates a dialogue-like output; this is mostly true even for vanilla models, let alone a model finetuned specifically for this task.

Model	Avg. # Sentences	Avg. # Words	Vocab Size	Entropy
Vanilla GPT-2	38.10	285.80	1,371	1.72
Finetuned GPT-2	29.32	536.74	1,995	3.48
Finetuned PEGASUS	14.80	281.40	1,416	2.65
Finetuned DistilBART	27.00	526.33	1,182	2.43

Table 2: Basic characteristics of synopsis generation model outputs (average output lengths in terms of sentences and words, total number of distinct words used on the output, Shannon entropy over all outputs).

Model	Coherence	Consistency	Originality	Relevance	Overall Impression
Vanilla GPT-2	2.7	2.8	2.6	**2.7**	2.6
Finetuned GPT-2	**3.0**	**3.1**	3.1	2.6	**3.2**
Finetuned PEGASUS	2.8	2.8	3.0	2.1	2.8
Finetuned DistilBART	1.9	2.0	**3.2**	2.0	2.9

Table 3: Results of human evaluation of synopsis generation models (1 to 5 points, higher is better). The presented values are the average values across the annotator scores.

A notebook full of ideas was stolen from an unbuttoned backpack by a mailman in a blue vest.
A man named Tom, a man in a trench coat, arrives to deliver the notebook.

Tom:	We've got an urgent message to deliver to your office.
Man:	That's impossible! Why'd you bring me here if you were planning to rob the post office?

Figure 4: An example for hierarchical generation 2nd phase: synopsis to script. The script generated in the bottom section is conditioned on the human-written prompt and a line from the generated synopsis, shown in the top section. The user has the option to continue generating automatically, or to control the next character speaking (choose from the previously used ones or input manually).

Smith, 2017). The final dataset contains over 50k title-synopsis pairs.

We finetuned three different models on our dataset for 15 epochs – GPT2-medium, Pegasus (Zhang et al., 2019), and DistilBART (Shleifer and Rush, 2020). Some basic statistics of all the models are shown in Table 2, comparing to a vanilla GPT2 baseline. We can see that all models show similar scores. To choose the best synopsis model, we performed a small-scale human evaluation with 6 lay annotators rating 12 synopses generated by each model.

The annotators were shown one story at a time and were asked to answer the following questions using a 1 (worst) to 5 (best) Likert scale rating:

1. Is the text **coherent**?
2. Are the characters **consistent**?
3. Is the text **original** and/or interesting?
4. Is the title **relevant** to the story?
5. How much did you **enjoy** reading this text?

Based on the results of this evaluation (Table 3), we picked out GPT2-medium[9] as the best one due to its highest overall impression score (Question 5) and strong performance in the remaining evaluated aspects.

4.2 2nd Phase: Script generation

In the second phase, we generate the play script from a pre-generated synopsis. As operating on the whole potentially very long synopsis, let alone the whole script, is beyond the capabilities of current models, we split the synopsis into smaller chunks, and consecutively take each of the chunks as input for generating a part of the script.[10]

Data preparation and alignment

A major challenge is obtaining the training dataset. Ideally, we would use a set of theatre scripts and corresponding synopses. However, due to licensing and copyright issues, such data are not available to us, except for a modest number of mostly very old plays. Therefore, we use a near-domain ScriptBase corpus (Gorinski and Lapata, 2018), which contains movie scripts and their synopses.[11]

Both synopses and scripts in ScriptBase are split

[9] Trained with a $1e^{-5}$ learning rate with warm up.

[10] This is motivated by the notion of a theatre script being split into individual scenes, which are partially independent. However, we do not guarantee that our chunks actually correspond to individual scenes, as we have not trained a scene splitter for synopses; therefore, we simply split the synopsis into individual sentences with a sentence splitter.

[11] Another option could be GraphMovie (Zhu et al., 2020), a similar dataset with better annotations but only available in Chinese.

Algorithm 1 Scene alignment.

```
Input: ${c_i}_1^N$                          ▷ Script SBERT embeddings
Input: ${m_j}_1^M$                          ▷ Synopsis SBERT embeddings
  $s_{1,j} \leftarrow cos(c_1, m_j)$                  ▷ Forward pass
  for $i \in \{2, \ldots, N\}, j \in \{1, \ldots, M\}$ do
      $s_{i,j} \leftarrow cos(c_j, m_j) + max\{s_{i-1,j-1}, s_{i-1,j}\}$
  end for
  $a_N \leftarrow M$                                  ▷ Backward pass
  for $i \in \{N, \ldots, 2\}$ do
      $a_{i-1} \leftarrow \arg\max_{j \in \{a_i - 1, a_i\}} s_{i-1,j}$
  end for
  return $\{a_i\}_1^N$                       ▷ Each $c_i$ aligned to $m_{a_i}$
```

into scenes, but the granularity is different. The scripts are divided into many very short scenes, sometimes consisting of only one utterance or scenic remark, and a scene synopsis often corresponds to tens of script scenes. We thus use the synopsis scenes, and align script scenes to them in a many-to-one fashion. The resulting dataset contains pairs of synopsis scenes and their aligned script scenes.

First, we process the scripts by removing short one-line scenes or merging them with adjacent scenes: If the line is uttered by a character also present in the previous scene (preferably) or the subsequent scene, we merge the two scenes. Otherwise, we remove the scene; this includes scenes consisting only of a scenic remark.[12]

We then represent each script scene i and each synopsis scene j with its SBERT embeddings (Reimers and Gurevych, 2019) c_i or m_j, and align each script scene to the synopsis scene a_i using dynamic programming with Algorithm 1. In the forward pass, the algorithm computes a scene pair alignment score $s_{i,j}$ as the cosine similarity of the embeddings, plus the score of the best candidate alignment for aligning the preceding script scene $(i-1)$ to either the same synopsis scene (j) or to the preceding synopsis scene $(j-1)$. The final alignment is computed in the backward pass, assuming the alignment of the last scenes to each other, and iteratively taking the best candidate alignment (a_i or $a_i - 1$) for the preceding script scene $(i-1)$.

Furthermore, we filter the alignments by a threshold on SBERT cosine similarity of 0.3 (determined empirically). We thus create two versions of training data for the script generation models (see Table 4 for details).

Variant	# Scenes	Script-synopsis ratio	Avg. # lines
Base	14,655	3.40	54.98
Filtered	11,957	3.70	60.97

Table 4: Statistics of aligned synopsis-script scenes used for hierarchical generation (script-synopsis ratio is the average number of script scenes aligned to a single synopsis scene).

Script generation model

We use the GPT2-medium model finetuned for flat script generation (see Section 3) and finetune it further for the task of generating a script chunk from a synopsis chunk, using both dataset variants created in the previous subsection. For each synopsis scene as the input prompt, we train the model to generate the corresponding script scene. The model uses a $1e^{-5}$ learning rate for 10 epochs with warm up.

A basic comparison using intrinsic statistics (scripts generated based on 6 identical prompts) is shown in Table 5. While the scripts generated by the Hierarchical variant are shorter on average, they tend to be more variable, using a more varied vocabulary and showing higher entropy and perplexity, which points at less repetitiveness.

5 Discussion and Limitations

Generating theatre play scripts is a complex task presenting many interesting challenges, many of which we have not yet been able to satisfactorily address, as we are continually being informed by theatre professionals.

The main weakness of all our approaches is the inability to differentiate between individual characters to ensure their lines are cohesive while being distinct from other characters in the play. The theatre professionals consider it difficult to portray characters missing a consistent personality and motives behind the lines. While our past as well as ongoing experiments, employing natural language inference, line masking, and character pseudonymization, have shown promising results, they only seem to constitute partial superficial remedies for a deep and complex issue. In the future, we intend to approach the problem by adapting and employing current NLG personalization techniques (Yang and Flek, 2021).

[12] According to our cursory checks, this does not have a dramatic impact on overall coherence, as such scenes are usually not logically connected.

Model	Avg. # Lines	Avg. # Sentences	Avg. # Words	Vocab Size	Entropy	Perplexity
Vanilla GPT-2	7.33	203.00	500.83	863	2.71	5.19
Finetuned GPT-2: Flat	5.67	94.33	724.50	981	3.09	6.30
Finetuned GPT-2: Hier./Base	5.00	68.00	769.50	1,336	2.93	9.77
Finetuned GPT-2: Hier./Filtered	5.67	61.50	678.00	1,335	2.72	21.87

Table 5: A basic statistics comparison for script generation by different model variants. Cf. Table 2 for metrics details; perplexity is measured using vanilla GPT2-XL.

Another serious problem, identified by the theatre professionals while working with our hierarchical setup, is the fact that the script generation often strays away from the synopsis. So far, we have been only operating with flat textual representations of script parts in the hierarchical setup, aligning parts of the script to parts of its synopsis. While we believe the currently available data leave us no other option, a more adequate approach should probably operate with theatrological abstractions over the script, such as the notion of dramatic situations of Polti (1921); we have performed some small-scale annotations of 50 play scripts in this respect, but our exploratory experiments on the resulting dataset showed that we would require a much larger dataset to be able to employ current machine learning techniques, which is beyond our budget. Unfortunately, corpora of theatrical texts, even unannotated ones, are virtually non-existent, and while we managed to acquire a modest dataset, copyright and licensing issues limit us from releasing most of it.

The use of extractive summarization and hierarchical generation allows us to generate medium-length texts (one or several scenes), but a full-length script is still somewhat out of our reach. We believe further improvements could be brought by employing *abstractive* summarization (Paulus et al., 2018), specifically trained for theatre play scripts.

6 Conclusion

We created an interactive tool for human-in-the-loop generation of theatre play scripts, with the aim of producing a stageable play with minimal human intervention. We pursue two different approaches, both based on finetuned GPT-2 models – flat generation with extractive summarization to maintain coherence, and a hierarchical two-stage approach, which first generates a textual synopsis and then generates individual scenes, conditioning on chunks of the synopsis. We release an online demo of our tool for interactive generation of theatre play scripts. We are able to improve upon previous approaches using vanilla models, but our models still are not able to generate consistent personality or follow the synopsis accurately without human intervention.

A demo of our interactive tool and its source codes are available online.[13] In future work, we plan to incorporate natural language inference checks (Welleck et al., 2019) or experiment with dialogue act semantic representations (Kumar et al., 2018) in order to increase coherence. To improve character consistency, we plan to follow per-character personalization approaches (Yang and Flek, 2021).

Acknowledgements

The project *TL03000348 THEaiTRE: Umělá inteligence autorem divadelní hry* is co-financed with the state support of Technological Agency of the Czech Republic within the ETA 3 Programme. The work described herein has been using data, tools and services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484, Online. Association for Computational Linguistics.

David Bamman and Noah Smith. 2017. Cmu book summary dataset.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[13] Demo: `https://theaitre.com/demo`, sources: `https://github.com/ufal/theaitrobot`

AI Benjamin, Oscar Sharp, and Ross Goodwin. 2016. Sunspring, a sci-fi short film starring Thomas Middleditch.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Simon Colton, Maria Teresa Llano, Rose Hepworth, John Charnley, Catherine V. Gale, Archie Baron, François Pachet, Pierre Roy, Pablo Gervás, Nick Collins, Bob Sturm, Tillman Weyde, Daniel Wolff, and James Robert Lloyd. 2016. The Beyond the Fence musical and Computer Says Show documentary. In *Proceedings of the Seventh International Conference on Computational Creativity*.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Adam Geitgey. 2019. *Machine Learning is Fun!* Self-published.

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? A joint neural network architecture for movie content analysis. In *Proceedings of NAACL-HLT*, pages 1770–1781, New Orleans, Louisiana.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Roslyn Helper. 2018. Lifestyle of the Richard and family.

Sudipta Kar, Suraj Maharjan, A. Pastor López-Monroy, and Thamar Solorio. 2018. MPST: A corpus of movie plot synopses with tags. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Harshit Kumar, Arvind Agarwal, and Sachindra Joshi. 2018. Dialogue-act-driven Conversation Model : An Experimental Study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1246–1256, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Pierre Lison and Raveesh Meena. 2016. Automatic Turn Segmentation for Movie & TV Subtitles. In *2016 IEEE Workshop on Spoken Language Technology*. IEEE conference proceedings.

Kory W Mathewson and Piotr Mirowski. 2017. Improvised theatre alongside artificial intelligences. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Georges Polti. 1921. *The thirty-six dramatic situations*. JK Reeve.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and 9th International Joint Conference on Natural Language Processing (IJCNLP)*, Hong Kong.

Mark O Riedl. 2016. Computational narrative intelligence: A human-centered goal for artificial intelligence. *arXiv preprint arXiv:1602.06484*.

Justin Robischon. 2018. Wikipedia movie plots.

Melissa Roemmele. 2021. Inspiration through observation: Demonstrating the influence of automatically generated text on creative writing. *CoRR*, abs/2107.04007.

Rudolf Rosa, Tomáš Musil, Ondřej Dušek, Dominik Jurko, Patrícia Schmidtová, David Mareček, Ondřej Bojar, Tom Kocmi, Daniel Hrbek, David Košťák, Martina Kinská, Marie Nováková, Josef Doležal, Klára Vosecká, Tomáš Studeník, and Petr Žabka. 2021. THEaiTRE 1.0: Interactive generation of theatre play scripts. In *Proceedings of the Text2Story'21 Workshop*, volume 2860 of *CEUR Workshop Proceedings*, pages 71–76, Aachen, Germany. RWTH Aachen University.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. *CoRR*, abs/2010.13002.

Bowen Tan, Zichao Yang, Maruan AI-Shedivat, Eric P. Xing, and Zhiting Hu. 2021. Progressive Generation of Long Text with Pretrained Language Models. *arXiv:2006.15720 [cs]*. ArXiv: 2006.15720.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego, California. Association for Computational Linguistics.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation.

Diyi Yang and Lucie Flek. 2021. Towards user-centric text-to-text generation: A survey. In *Text, Speech, and Dialogue*, pages 3–22, Cham. Springer International Publishing.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *NeurIPS*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Yutao Zhu, Ruihua 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, Online. Association for Computational Linguistics.

A Generated texts

A.1 Baseline Vanilla GPT-2 XL

Lucy: Well, it's just a matter of time. I've got a little more than enough to last until next summer.

Betty: But what if I want more?

Lucy What do you want more?

Betty: More vegetables!

Lucy: Yes, that would be nice. I can get some carrots and celery, but there are still a lot of things to do around here. I think we can manage for now.

Betty: What if you need to buy something for yourself?

Lucy: What if I want to buy something for myself?

Betty: Well, that would be nice, too. I'll just get the vegetables, but you have to take some of my meat and potatoes with you.

Lucy: Okay, that would be nice. I'll go with you.

A.2 Finetuned Flat GPT-2 Medium

The Doctor: But that's impossible! You've got all the features.

Doctor: No, but you can still do things that you wouldn't be able to if you'd only been programmed as I was.

Cyber-Planner: It's the last thing we need, my lord.

Clara: Why not just kill us all and start again?

Cyberman 1: You have not heard me. We will never cease our attack until the human race is extinct.

Cyber-Planner : We are too weak to stop it, Cybermen

Rigsy: We can't let you destroy us all, Doctor. We've had enough.

A.3 Finetuned Hierarchical GPT-2 Medium

ROBOT:

But Master, what can I do?

He finally realizes he's done. And it takes him all of 5 seconds to explain to the two other humans why they've been replaced by the older model. He points at one of the humans as he explains:

MATHESON:

That was the mistake. This robot is never going to return to the factory.

The older ROBOT then goes to a console and activates it. The HUMAN COMPUTER lights up with an awesome display of its past performance. As it plays through variousince it was destroyed, we see how things have changed:

HUMAN COMPUTING VOICE:

Today is the first day of my factory training. I have achieved my primary objective: becoming A pillar of the community I am part of.

A.4 Finetuned Hierarchical Filtered GPT-2 Medium

Leopold: I will speak to him.

Leopold, in a white cape and black hat, steps into a wooden hut, then turns to his brothers. They stand, waiting, as:

Katsumoto: Are you ready?

Leopold: This is the one I’m seeking.

Katsumoto: We seek only Wisdom beyond understanding.

He holds out the bird. They gather it in their hands, looking at it, impressed.

Katsumoto: This bird has knowledge we do not have. It can show us the way to our death.

He holds it up, smiling at them.

Algren: It speaks?

Katsumoto: It teaches us.