THEaiTRE 1.0: Interactive Generation of Theatre Play Scripts

Rudolf Rosa[1], Tomáš Musil[1], Ondřej Dušek[1], Dominik Jurko[1], Patrícia Schmidtová[1], David Mareček[1], Ondřej Bojar[1], Tom Kocmi[1], Daniel Hrbek[2,3], David Košťák[2], Martina Kinská[2], Marie Nováková[1,2], Josef Doležal[3], Klára Vosecká[3], Tomáš Studeník[4], Petr Žabka[4]

[1]Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Prague, Czechia
uru@ufal.mff.cuni.cz

[2]The Švanda Theatre in Smíchov
Prague, Czechia
hrbek@svandovodivadlo.cz

[3]The Academy of Performing Arts in Prague, Theatre Faculty (DAMU)
Prague, Czechia

[4]CEE Hacks
Prague, Czechia
info@ceehacks.com

Abstract

We present the first version of a system for interactive generation of theatre play scripts. The system is based on a vanilla GPT-2 model with several adjustments, targeting specific issues we encountered in practice. We also list other issues we encountered but plan to only solve in a future version of the system. The presented system was used to generate a theatre play script premiered in February 2021.

1 Introduction

The THEaiTRE project[1] [RDK+20] aims to produce and stage the first computer-generated theatre play on the occasion of the 100th anniversary of Karel Čapek's play *R.U.R.* [Čap20], in which the word "robot" first appeared.

In this paper, we describe the THEaiTRobot 1.0 tool, which allows the user to interactively generate scripts for individual theatre play scenes. The tool is based on the GPT-2 XL [RWC+19] generative language model, using the model without any fine-tuning, as we found that with a prompt formatted as a part of a theatre play script, the model usually generates continuations that fit the format well. However, we encountered numerous

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Finlayson (eds.): Proceedings of the Text2Story'21 Workshop, Online, 1-April-2021, published at http://ceur-ws.org

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

problems when generating the script in this way. We managed to tackle some of the problems with various adjustments, but some of them remain to be solved in a future version.

Our tool was used to generate the script for a new play, *AI: Když robot píše hru* (*AI: When a robot writes a play*), which was premiered online on 26th February 2021.[2] Although there were various forms of human intervention when generating the script, we estimate that over 90% of the text comes from the automated tool; moreover, most of the interventions were similar to those a dramaturge and director would do in case of a human-written script (making cuts, rearranging lines, reassigning characters, minor edits of the lines, etc.).[3] We present a preliminary analysis of the interventions in Section 2.1. The GPT-2 model was found unfit for generating long and complex texts such as a full play script; we therefore generated several individual scenes and then a dramaturge joined them into a full play.

We have published a video showing the operation of THEaiTRobot 1.0, a sample of its outputs, and its source codes:

- Video: `https://youtu.be/ksrZouM7Wyg`
- Sample outputs: `http://bit.ly/theaitre-samples`
- Source codes: `http://hdl.handle.net/11234/1-3507` [RDK+21]

2 The Generation Process

The process of generating a theatre play scene script starts by the user (a theatre dramaturge in our case) defining the start of the scene, typically a setting and several initial lines of dialogue, which defines the theme of the scene, introduces the characters, and encourages the GPT-2 language model to start generating a dialogue. For the first play, we defined a set of inputs revolving around a common topic to ensure some basic coherence of the whole play.[4] The THEaiTRobot tool then uses the vanilla GPT-2 XL model to generate continuing lines, which then get translated from English to Czech by a Machine Translation service. The user has the option to discard any generated line (together with all subsequent lines), prompting the tool to generate a different continuation.[5] The user can also manually enter a line into the script, which becomes part of the input for GPT-2.[6] The tool itself is implemented as a web application with a server backend, using the Huggingface Transformers library [WDS+20].

2.1 Preliminary Analysis of Human Interventions

We are currently in the process of performing a detailed audit of the genesis of the final script of the first play, which we intend to publish once finished. So far, we have completed the analysis of the first scene, *Death*.

The first scene consists of 60 lines, out of which 45 were used without any changes from the generated script, while the remaining 15 lines were slightly modified. We detail the modifications in Table 1; note that in some lines there was more than one intervention. Also, 11 lines from the generated script were deleted.

The generation process was initiated with a prompt consisting of a scene setting and two character lines; only one of them also became part of the final scenario. In total, 91% of the words in the script of the first scene are used as they were generated, while 9% of the words were added, changed or reordered.

We have only performed the analysis on the English side of the script. On the Czech side of the script, there are additional edits which fix some errors of the automated machine translation, as explained in Section 2.3.4.[7]

[2] `https://www.svandovodivadlo.cz/inscenace/673/ai-kdyz-robot-pise-hru/3445`

[3] In fact, the dramaturge and the director reported that they made *fewer and smaller edits* to the script than they typically do with a human-written script.

[4] The play tells a story of a robot trying to find his place in the human society. Each scene revolves around a typically human theme, such as death, love, sex, or work, and the robot learns about this theme through the interaction with a human character.

[5] This option was used for approximately 5% of the lines in the script of the first play.

[6] This option was used very rarely. Apart from the input prompts, only approximately 1% of the lines were hand-written and manually entered into the script.

[7] On the Czech side, approximately 23% of words were modified, but most of the additional modifications were fixes of incorrect T-V distinction or gender.

Table 1: Audit of human interventions into the script of the first scene, *Death*.

Count	Intervention	Example (before – after)	
45	No intervention	Robot: I love you so much I want to hug you to death.	
2	Human-written prompt, not part of script	It's morning. Robot enters room of his master who is really old and sick. Robot sees that his master is not doing very well this morning. He sits at the edge of his bed and takes his hand.	
1	Human-written prompt, part of script	Master: We both know I am dying.	
9	Minor edit of language	Master: No. Don't say that. I want to **have an end**!	Master: No. Don't say that. I want to **enjoy my ending**!
5	Swap of character names	**Master**: You are going to die in your sleep.	**Robot**: You are going to die in your sleep.
3	Swap of perspective	Master: I don't think **I** could hug **you** to life.	Master: I don't think **you** could hug **me** to life.
1	Duplication of a generated line	Robot: I'm afraid of what I've been doing here.	**Master: I'm afraid of what I've been doing here.** Robot: I'm afraid of what I've been doing here.

2.2 Resolved Issues

2.2.1 Set of Characters

The model does not work with a limited set of characters naturally and tends to forget characters and invent new characters too often.[8] We resolve this by modifying the next token probability distribution within the GPT-2 model, so that at the start of a new line, only tokens corresponding to character names present in the input prompt are allowed. We also boost probabilities of characters that have not spoken for some time.[9]

2.2.2 Repetitiveness

GPT-2's generation may get stuck in a loop, generating one or several lines again and again. We managed to resolve this by modifying the hyperparameters of GPT-2, changing repetition penalty from 1.00 to 1.01. As a backup, we also automatically discard any generated repeated lines and prompt the model to generate another continuing line.

2.2.3 Limited Context

The variant of the GPT-2 model which we are using has a limit of 1024 subword tokens, within which both the input prompt and the generated output must fit. The typical solution is to crop the input at the beginning so that it fits into the window with sufficient space for generating the output. However, this means forgetting potentially important information from the input prompt and the previously generated text, which can lead to an unwanted continual topic drift and also to generating contradictory text; the text is still locally consistent, but as a whole it may be inconsistent.

To handle this issue, we introduce automated extractive summarization into the process, hoping that the summarization algorithm will identify the most important pieces of information to remember. Whenever the input for GPT-2 (the input prompt + the so far generated script) exceeds a preset limit of $M = 924$ tokens,[10]

[8] If there are only two characters in the scene, the model often keeps to them or only introduces 1-2 additional characters. If the number of characters is higher than 3, the model usually forgets some of them after some time. Also, some character names push the model in unintended directions. We have seen the model generalize over character names originally involving only "Robot 1" and "Robot 2", with the model continually introducing "Robot 3", "Robot 4", "Robot 5", etc. We have also observed the model to immediately change a character called "Vladimir" to "Vladimir Putin".

[9] Specifically, we multiply each character probability by 2^c where c is the number of lines for which the character has not spoken.

[10] Most script lines in our setting fit within 100 tokens, so ensuring there is space for generating at least 100 tokens means that usually the model will generate a complete line, ending with a newline symbol; in case the generated line is too long, it is simply cut off once the limit of 1024 tokens is depleted.

we summarize the input using TextRank[11] [MT04] before feeding the input into the GPT-2 model:

- We keep all lines within the last $R = 250$ tokens from the input[12] to ensure local consistency.
- We summarize all the preceding lines into $N = 5$ lines (while keeping their original order) to ensure global consistency.
- We concatenate the summary and the kept lines.
- If the resulting text is still longer than M tokens, we crop it at the beginning to M tokens.

2.2.4 Machine Translation

The GPT-2 model operates on English, while we want to generate a Czech script. We therefore automatically translate the generated script using the CUBBITT [PTT+20] neural translation model. As the translation tends to discard character names from the lines, we add them by identifying them in the input and translating them independently.

2.3 Unresolved Issues and Future Plans

2.3.1 Generating a Whole Play

The model is not able to generate a long and complex text such as a full theatre play script. To resolve this, we intend to generate the script hierarchically, first generating a synopsis for the whole play, then expanding it into synopses for individual scenes, and finally generating each scene individually based on its synopsis. This approach is inspired by the work of Fan et al. [FLD18, FLD19], who take a similar coarse-to-fine approach to story generation. Our situation is, however, more complex, as we plan to use one more step of the hierarchy.

2.3.2 Character Personalities

The characters in the play do not seem to have independent personalities in the generated script; the model seems to simply ensure consistency with already generated text, not taking the character names into account much. The character personalities thus appear to switch and merge. We intend to resolve this by learning theatre character embeddings and using them to condition the language model. We plan to resolve this by clustering our data into several basic character personality types [AKDM19], then train separate character-aware language models, either by finetuning the GPT-2 model, or by using adapter models [MIL+20, WTD+20].

2.3.3 Dramatic Situations

The text is generated word by word and line by line, whereas human authors of theatre plays typically operate on a more abstract level, such as dramatic situations [Pol21].[13] While there is some work on identifying dramatic turning points [PKL19, PKFL20], it is too coarse-grained for our application. We are thus currently annotating a corpus of theatre play scripts with a modified set of dramatic situations, and plan to enhance the tool with this abstraction, either by adding one more layer in the hierarchical setup, or by using special tokens or embeddings to mark dramatic situations in the generated text.

2.3.4 Machine Translation

The MT model we use is tuned for news text, not theatre scripts, and translates each sentence independently. This leads to various issues, including errors in morphological gender (which should pertain to the character), variance in the honorific T–V distinction (which may vary but should be consistent for each pair of characters), and erroneous sentence splitting. We intend to tackle these issues by using a document-level translation system which takes larger context into account, fine-tuning the model on a corpus of theatre play scripts, and adding various heuristic modifications where necessary.

[11] We use the `pytextrank` library with minor modifications to reflect the specific structure of our inputs, so that the algorithm returns N most important (potentially multi-sentence) *full lines* from the script instead of just N most important *sentences*. We set `limit_phrases=100`.

[12] We find the first newline symbol in the last R tokens and keep all the lines after it.

[13] `https://en.wikipedia.org/wiki/The_Thirty-Six_Dramatic_Situations`

2.3.5 Evaluation

We have not yet devised any automated or semi-automated evaluation setup to measure the quality of the generated scripts. Our design decisions so far have thus been based solely on manual analyses of small numbers of outputs, performed by theatre experts. While such analyses are very trustworthy, they are not easy to perform at an adequate scale. On the other hand, we are not aware of any meaningful automated measures of theatre script quality. We are currently exploring automated chatbot quality measures, which might or might not provide some useful indications of the script quality. We are also working on making the manual evaluation more efficient by designing a set of evaluation prompts and a standardized binary evaluation procedure.

Nevertheless, the ultimate evaluation of a theatre play always is the reception by the audience and the critiques; which, in case of the presented play, have been rather positive.[14]

3 Conclusion

We have developed THEaiTRobot 1.0, a tool for interactively generating theatre play scripts. The tools is based on GPT-2, with several modifications targeting encountered issues. We have also discussed persisting issues and suggested remedies for a future version.

We used the tool to create the first predominantly machine-generated theatre play script, which premiered on 26th February 2021. Another play, to be generated by an improved version of the tool, is planned for 2022.

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 ÉTA 3 Programme.

References

[AKDM19] Mahmoud Azab, Noriyuki Kojima, Jia Deng, and Rada Mihalcea. Representing Movie Characters in Dialogues. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 99–109, Hong Kong, November 2019.

[Čap20] Karel Čapek. *R.U.R. (Rossum's Universal Robots)*. Aventinum, Ot. Štorch-Marien, Praha, 1920.

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, New Orleans, LA, USA, June 2018. arXiv: 1805.04833.

[FLD19] Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for Structuring Story Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy, July 2019. Association for Computational Linguistics.

[MIL+20] Andrea Madotto, Etsuko Ishii, Zhaojiang Lin, Sumanth Dathathri, and Pascale Fung. Plug-and-Play Conversational Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2422–2433, Online, November 2020. Association for Computational Linguistics.

[MT04] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[PKFL20] Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. Screenplay Summarization Using Latent Narrative Structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1920–1933, Online, July 2020.

[PKL19] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie Plot Analysis via Turning Point Identification. 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 1707–1717, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] https://tinyurl.com/yccsteot

[Pol21] Georges Polti. *The thirty-six dramatic situations*. JK Reeve, 1921.

[PTT+20] Martin Popel, Marketa Tomková, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15, 2020.

[RDK+20] Rudolf Rosa, Ondřej Dušek, Tom Kocmi, David Mareček, Tomáš Musil, Patrícia Schmidtová, Dominik Jurko, Ondřej Bojar, Daniel Hrbek, David Košťák, Martina Kinská, Josef Doležal, and Klára Vosecká. THEaiTRE: Artificial intelligence to write a theatre play. In Alípio Jorge, Ricardo Campos, Adam Jatowt, and Akiko Aizawa, editors, *Proceedings of AI4Narratives — Workshop on Artificial Intelligence for Narratives*, volume 2794 of *CEUR Workshop Proceedings*, pages 9–13, Aachen, Germany, 2020. RWTH Aachen University, RWTH Aachen University.

[RDK+21] Rudolf Rosa, Ondřej Dušek, Tom Kocmi, David Mareček, Tomáš Musil, Patrícia Schmidtová, Dominik Jurko, Ondřej Bojar, Daniel Hrbek, David Košťák, Martina Kinská, Marie Nováková, Josef Doležal, and Klára Vosecká. THEaiTRobot 1.0, 2021. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, February 2019.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[WTD+20] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. *arXiv:2002.01808 [cs]*, December 2020.