Survey: Automatic Movie Plot And Script Generation

Vishal Pramnik
vishalpramanik35@gmail.com

Abstract

Automatic story generation has a long-standing tradition in the field of Artificial Intelligence. The ability to create stories on demand holds great potential for entertainment and education. For example, modern computer games are becoming more immersive, containing multiple story lines and hundreds of characters. This has substantially increased the amount of work required to produce each game. However, by allowing the game to write its own story line, it can remain engaging to the player whilst shifting the burden of writing away from the game's developers. In education, intelligent tutoring systems can potentially provide students with instant feedback and suggestions of how to write their own stories. Although several approaches have been introduced in the past (e.g., story grammars, story schema and autonomous agents), they all rely heavily on handwritten resources. Which places severe limitations on its scalability and usage. In this report, I have explored a specific area of Natural Language Generation: automatic movie plot and script generation. The time taken for a script generation by a human writer may take over a year. My target is to reduce this time as much as possible and help the script writer with new ideas and events for the plot and the script.

1 Introduction

Natural Language Generation (NLG) is a branch of Artificial Intelligence (AI) that generates language as an output on the basis of data as input.

There has been a significant rise in the adoption of NLG into business, in recent times. As it turns out, a machine can articulately communicate ideas from data at remarkable scale and accuracy. When a machine automates the regular mundane routine analysis and communication tasks, productivity increases and employees can focus on decision-making and end actions.

According to a report by Gartner, it is predicted that by 2022, 25 percent of enterprises will use some form of natural language technology. The goal of NLG systems should be to understand how to best communicate what it knows. For that, it needs to have an unbiased and clear picture of the world rather than random strings of text. Simple NLG systems are capable of taking in ideas in the form of data and transforming them into language.

Apple's Siri uses this concept of linking ideas to sentences to in turn produce a limited yet succinct response. Another real-world example of NLG is robot journalism, which automates the process of reporting and content writing by producing comprehensive articles written in plain natural language, based on input data.

There are six stages to an automated text generating process. For the sake of clarity, we'll describe each level using the following example of a robot journalist reporting on a football match:

- **Content Determination:** The content's boundaries should be established. Often, the data contains more information than is required. For example, in football news, content about goals, cards, and penalties will be valuable to readers.

- **Content Determination:** The data that has been evaluated is then interpreted. Patterns in processed data can be recognised using machine learning algorithms. This is where the data is placed in its proper context. This stage, for example, identifies information such as the match winner, goal scorers assisters, and the minutes when goals are scored.

- **Sentence Aggregation:** This method is also known as micro planning, and it entails selecting the expressions and words in each sentence for the end-user. To put it another way, this is the step at which distinct sentences are grouped together in context based on their relevance. The first two sentences, for example, have different meanings. If, on the other hand, the second occurrence occurs just before half-time, these two phrases can be combined to form the third sentence.

- **Grammar Check** The grammaticalization stage ensures that the entire report is written correctly in terms of grammatical structure, spelling, and punctuation. This includes validating actual text against syntax, morphology, and spelling norms. Football games, for example, are written in the past tense.
- **Language Implementation:** This stage involves inputting data into templates and ensuring that the document is output in the right format and according to the preferences of the user.

On various aspects of computational storytelling, numerous surveys have been written. Using measures like the novelty of the stories and the degree to which users are engaged in the storytelling process, (Gervás, 2009) gives a chronological overview of storytelling systems with a focus on computational creativity. Interacting with a computer system to create stories is the topic of (Riedl and Bulitko, 2013) study on interactive intelligence, a type of digital interactive storytelling. The research paper touches on developing storylines and creating characters. (Riedl and Bulitko, 2013) talks about human-centered computational narrative and how it might enhance applications of artificial intelligence. The Paper provided some insight into the problems with common sense reasoning and story development in machine learning. However, as it is not its purpose to do so, it does not elaborate on these difficulties.

2 Problem definition

The goal of this project is to create a tool that shortens the time needed to create a movie. A script writer's assistant has to be made. Plot and script generation have been the two key areas of concentration. This tool will also help the script writers with new and creative ideas for movies. The plot generation model creates a movie plot of approximately 700 words from a brief narrative description (between 25 and 100 words). The scene generation model produces a scene with the script's structure preserved from a brief input of two to three sentences.

3 Motivation

The ability to generate stories automatically is most relevant in education where stories are used to help teach children. Currently, however, story creation relies on human authors and the writing process demands considerable effort and time to ensure stories are both informative and entertaining. Teaching resources are therefore limited to the stories that are in circulation. Computational story writing has the potential to greatly increase the number of stories available by removing the reliance on human authors. Also, current research of interactive tutoring systems has highlighted the ability to provide students with feedback, specifically tailored to meet their needs based on automatic assessment of their work. The ability to analyse a story in development,offering suggestions and highlighting areas of concern would be invaluable to young authors. As a prototype for such a system, StoryStation (Halpin et al., 2004) is able to discern the difference between good and poor attempts by students at a story retelling task. Automatic storytelling also holds great potential as a component in interactive environments which can be used as learning aids. (Robertson and Good, 2003) propose GhostWriter, a system that allows children to interact in a story environment.

As technological media becomes more pervasive in human culture, (Murray and Murray, 2017) foresees a greater role for digital storytelling in everyday life. The more we interact with the web, social networking and computer games, the more we expect to find narrative structures to make them more accessible and entertaining. Stories are already used extensively in the entertainment industry, for instance within computer games. Many computer games, such as Role Playing Games (RPGs), allow the gamer to assume the role of a character and participate in a story line throughout the game. As the performance and processing power of games consoles increase, so does the potential for larger and more sophisticated gaming environments. Recent games (e.g., The Elder Scrolls IV: Oblivion and Fallout 3) include multiple story lines, hundreds of NPCs (Non Player Characters), game locations and thousands of lines of dialogue, resulting in games that involve hundreds of hours of game-play. Clearly, games of this magnitude require substantial amounts of work in developing. Also, the majority of these games will have their plots set during development and as a result each consecutive play of the game will be less interesting.

4 Background: Terminology and Definitions

4.1 Study of the structure of a Movie Plot

The plot is what happens in a movie, i.e. the narrative sequence of events that determines the characters' fate. As a result, movie plots are the events that occur in a specific order in order to demonstrate cause and effect. Because the plot is what happens in a story, it is required. Nothing would happen if there was no plot, and there would be no story. The plot is what happens in a movie, while the tale explains us why and how something happened. Stories help us structure and understand the world around us, and we learn about others and ourselves by following a series of events (i.e. plot) and how events effect character change. Great storylines are satisfying to us for two reasons: one plot point flows logically to the next while simultaneously surprise and delighting the viewer.

Great plots depict characters making decisions and the consequences of those decisions, generating a chain reaction that raises the stakes with each new decision. Plants and payoffs are also crucial elements in great plots, as plot points that appear inconsequential early on return with a vengeance near the finish. The most crucial thing for a plot to do is feel meaningful; for a plot to seem satisfactory, it must eventually arrive at some form of point, whether it's a thematic occurrence, a character growth moment, emotional catharsis, or tension relief. Great plots don't have to have happy endings to be satisfying, but they do need to feel closed up.The skeleton backbone of a storyline that shapes the story is best defined as story structure.

Instead of a random sequence of events impacting a random sequence of characters indefinitely over two hours, story structure directs particular plot points to occur at specific periods in the story, forming a narrative arc.For determining the plot structure of your story, you can use a variety of framing approaches.The three-act framework, which divides your story into a beginning, middle, and end, is the most popular framing method. The five-act form, which Shakespeare utilised in most of his plays, is another well-known structural device.

You can employ structure to further define a story's ebbs and flows with a variety of structural framing approaches, regardless of how many "acts" it has. Some of the most well-known screenwriters include:

- The eight-sequence approach divides your story into eight mini-arcs known as sequences.
- The save the cat approach entails honing your story to adhere to a very particular, page-by-page beat sheet for when each plot point in a script should occur.

Storytelling has been one of the oldest means of communication for humans. Stories differ from other kinds of texts in many ways. Stories are meant to be coherent, and good stories are those which also do not have any loose ends. In order to ensure that the stories are good or worthwhile, people have created various templates that help in storytelling.

The most famous template of a story is the 3-act structure. The setup, the confrontation and the resolution make up the three acts. Generally, the second act is the largest and twice the size of the other two acts. Although this template has been used throughout the years, it was primarily popularized by Syd Field in his 1979 book *Screenplay: The Foundation of Screenwriting*.

There have been a number of templates created to assist with story writing. Some of them are extensions of the 3 act structure while some of them are made with other themes deciding the structure. One of the famous themes to create structure in stories is to focus on the protagonist.

There are a lot of mythologies and religious texts from ancient civilisations and some of them revolve around a protagonist. Ramayana revolves around Lord Ram and Mahabharata mainly revolves around Lord Krishna. These myths or legends are examples of stories whose theme is built around the protagonist.

4.1.1 The Hero's Journey

In the 20th century, the rise of cinema led to a breakthrough in the storytelling theme. (**?**) introduced *The Hero's Journey* which is a story template with 17 steps within a protagonist's journey. The template mainly consists of 3 divisions:

1. Departure
2. Initiation
3. Return

These 3 acts are further divided to a total of 17 steps. The *Departure* act shows the protagonist live in his ordinary world and due to some circumstances he receives a call to go on an adventure. The *Initiation* act shows the protagonist travel to the new world and face new challenges. He then reaches the lowest point of the journey and must overcome the main enemy or challenge. The *Return* act shows the hero return to his ordinary world again. This time he is a changed person and has a reward of kind with him. He finally gains spiritual power over both worlds.

4.1.2 The Beat Sheet

The Hero's Journey inspired many more themes that revolved around a protagonist. Blake Snyder released his book SAve The Cat in 2005 (Snyder, 2005), and introduced the concept of a Beatsheet to write scripts. The Beatsheet is an ordered template that allows a scriptwriter to focus on particular twists and concepts of the protagonist's journey.

Snyder starts by explaining the Beatsheet in 15 steps and then expanding it from 15 to 40 to cover all the scenes of a script. Since a 2 hour movie has around 110 scenes, each beat needs to cover only 3 scenes. This makes it easier for the scriptwriter to work on different areas of the script and then combining it to complete the entire script. In the table below, is the 15-part Beat Sheet divided into 3 acts.

4.2 Study of the structure of a Movie script

A screenplay, or script, is a written work by screenwriters for a film, television program, or video game. These screenplays can be original works or adaptations from existing pieces of writing. In them, the movement, actions, expressions, and dialogues of the characters are also narrated. A screenplay written for television is also known as a teleplay.

There are several ways to structure a screenplay. The classic structure is to divide a screenplay into three acts: the set-up, conflict, and resolution. Countless stories adhere to this format, and there's a reason why it has been the go-to structure for films pretty much since cinematography began.A movie script is made up of multiple inter-related scenes.

A movie script, also known as a screenplay is a document that ranges anywhere from 70-180 pages. Most movie scripts come in around 110 pages, but there are a number of factors that play into the length. The basics of script formatting are as follows:

- 12-point Courier font size
- 1.5 inch margin on the left of the page
- 1 inch margin on the right of the page
- 1 inch on the of the top and bottom of the page
- Each page should have approximately 55 lines
- The dialogue block starts 2.5 inches from the left side of the page
- Character names must have uppercase letters and be positioned starting 3.7 inches from the left side of the page
- Page numbers are positioned in the top right corner with a 0.5 inch margin from the top of the page. The first page shall not be numbered, and each number is followed by a period.

A particular scene in a script contains the following components.

- **Screen heading / sluglines** - The scene heading is there to help break up physical spaces and give the reader and production team an idea of the story's geography. Either INT. for interior spaces or EXT. for exterior spaces is chosen. Then a description of the setting, and then the time of day.There are rare cases where the scene begins inside and goes outside, or vice versa, and in these situations you may write INT/EXT. or EXT/INT. Some scripts take place all around the world, so often screenwriters will use multiple hyphens to give the scene headings even more detail. This helps the screenwriter avoid having to point out the geographical location in the action lines, saving space to write more about the actual story and keep readers engaged in the story, not the formatting.
- **Subheading** - Often, writers will use subheadings to show a change in location without breaking the scene, even if the scene has shifted from INT. to EXT.It is assumed that readers will understand the change in space while retaining the idea that the time of day is the same - even continuous.The reason many writers do this is to avoid the notion that we

Act 1	Act 2	Act 3
Opening Image Theme Stated Setup Catalyst Debate	Break into Act 2 B Story Fun and Games Midpoint Bad Guys Close In All is Lost Dark Night of the Soul	Break into Act 3 Finale Final Image

Table 1: The 15-part Beat Sheet under 3 Acts

have entered an entirely new scene, though you could always include CONTINUOUS in place of DAY or NIGHT by creating an entirely new scene heading.It's a matter of personal style and rhythm vs. production considerations.

- **Action lines**-This comes just after the Scene Heading. The action lines tells what is happening in the scene.Action lines are where we describe the visual and audible actions that take place on screen. We want to write in third person in present tense.

- **Character names**-When we introduce a character in a screenplay, we want to use all-capital letters for the name of the character, then a reference to their age, and finally some information about their traits and personality. Again, screenwriters have found other ways to do this, but this is the most common and production friendly way to introduce a character. Sometimes to introduce characters in a screenplay that goes into the creative considerations of introducing characters.

- **Dialogues** - of the character.Our lines of dialogue will be set underneath the character to which they are assigned. Dialogue is pretty straightforward from a formatting standpoint, but it is the most difficult part of screenwriting.

- **Extensions** - occur along with the character name that provides extra information about the character or the dialogue of the character.Eg:V.O(Voice Over),O.S(Off Screen) etc.

- **Parenthetical** - You can use a parenthetical inside your dialogue to show small actions, or even a change in mood without having to jump out to an action line. Parentheticals are really good for directing actors, and adding sarcasm and nuance to performances on the page.

- **Transitions**-In the bottom right of the page Transition is placed , but in modern screenwriting these seem to be used less and less. The transitions that seems to have really stood the test of time are CUT TO: and FADE OUT.

5 Literature Survey

The Automatic Story Generation can be broadly classified into two categories which are: Symbolic Approach and the Neural Approach. This chapter discusses the two approaches.

5.1 Symbolic Approaches

In this approach the activity of automatic story generation is carried out using symbolic guidelines which are given by the user as input to the model . One of the main strengths of this approach is the ease with which it is possible to delimit the path followed by the actions of the story. However, one of its disadvantages is the lack of creativity, in other words, stories tend to be very repetitive and have a very reduced repertoire of events. These works can be divided into two main areas, case-based and event-based

5.1.1 Case-based Methods

Case-based methods are so-called because they are strongly restricted to solving a particular theme, either the stories domain, the author goals, or both.The new stories hence created are based on the previous stories that are stored in its database by the user.

5.1.2 Event-based Methods

The event-based methods use a concept called events to construct the story-line. The events are the actions or the transitions from one state to another.

5.2 Neural Approaches

Every big advancement in the neural networks field has been used for automatic story generation. Some of the proposals for generating text have been sequence-by-sequence models (seq2seq) with recurrent neural networks (RNN) or GRUs or LSTMs. Another proposal is unsupervised learning utilizing Generative Adversarial Networks (GAN) or Bidirectional Associative Memories. Although neural-based models solve the lack of novelty issues found in the symbolic systems due to their unstructured generation, this advance comes at the cost of less controllability and plot coherence. In this section, we shed light on a few approaches to the problem of controllability, discuss their strengths and weaknesses, and compare their methodologies.

5.3 Related Works in Text Generation

The following are the works done on text generation.

5.3.1 Symbolic Approaches

The paper (Meehan, 1977) generates different character portrayals and their respective goals. The system must look for an outcome for the resolution of these goals using an inference engine based on common sense reasoning theories; all the stories are based on King Arthur's stories. In (McCoy et al., 2014) the authors generate prom stories using knowledge bases about the social structure, its norms, the cultural aspect, and the desires of the characters, as well as concepts of social interaction.In León and Gervás (2014), the authors use states of the story-world and simulate the interactions between the world and the different characters, choosing the simulation that best corresponds to the established narrative characteristics.

5.3.2 Neural Approaches

The following are the works using the neural approaches.

5.3.3 Reinforcement Learning

(Tambwekar et al., 2019) aimed at controlling the story plot by controlling its ending and events order. They proposed a deep reinforce approach to controlled story generation with a reward shaping technique to optimize the pre-trained sequence-to-sequence model (Martin et al., 2018). Their reward function encompasses two main parts, the distance to the goal verb and the story verb frequency. The distance to the goal verb measures how many lines between a generated verb and the goal verb in training stories. Simultaneously, the story verb frequency counts the stories with both the goal verb and the generated verb. They evaluated their model on plot coherence and goal achievement, length, and perplexity. Their method was better than their base model alone in the aspects being assessed. However, this approach requires training the model for every new goal, which can be inconvenient for the users. Another drawback to this model is it uses the sequence to sequence model (Martin et al., 2018), which generates stories as sequences of objects

5.3.4 Fusion Model

(Fan et al., 2018) attempts to solve the plot controllability problem by dividing the generation process into two levels of hierarchy: a premise, and a story. The premise provides an overall sketch of the story, which will be utilized to write the story. This fusion model combines a convolutional sequence-to-sequence model with a self-attention mechanism to improve generated story quality. A convolutional network first generates a writing prompt which then, becomes the input to the sequence-to-sequence model and guides it in generating a story conditioned on the prompt. Their model was superior in both human evaluations and perplexity scores than a traditional sequence-to-sequence method. Conditioning on the generated premise makes the generated story plot consistent and has an improved long-term dependency. Overall, this approach improves the shortcomings of the previous work by writing the stories directly and being conditioned for different prompts without retraining. Yet this model also has its own limitations. First, it relies heavily on random sampling for the generation, which are prone to errors. Second, it suffers from text repetition in the generated stories. Lastly, the generated prompts are generic and less interesting than human written writing prompts, which often generate boring stories.

5.3.5 Plan and Write

(Yao et al., 2019) proposed the Plan-and-write story generation framework. The authors leveraged some of the characteristics of symbolic planning and integrated it into a neural system. Their work improves the previous literature in that it uses the titles to generate controlled storylines rather than the auto-generated writing prompts directly. They utilize storyline planning to improve the generated stories' quality and coherence and thus control the generation. They explore several story planning strategies to see their effect on story generation. This framework takes as an input the title of the story and then generates a storyline. The storyline and the title are then used as input to control the story generation in a sequence to sequence model. They also proposed two metrics to evaluate their model, inter-story repetition, and intra-story repetition. The evaluations showed that the model is more superior to the used conditional language model baselines. Those evaluations also showed that the model suffers from several major problems: repetition, going off-topic, and logical inconsistencies. It also utilizes a sequential language model to approximate the story plot, which simplifies the structure and depth of a good story plot, suggesting that generating coherent and logical story plots is still far from being solved.

5.3.6 Plot Generation

(Rashkin et al., 2020) proposed a transformer-language-model-based system that generates multi-paragraph stories conditioned on specified outlines for these stories. This model shows improvements in the narrative over the previous work. The approach utilizes memory state tracking and discourse structures to better control the generated story plot and keep track of the generated lines to maintain coherence. The outlines are represented with an unordered list of high-level, multi-word descriptions of events occurring in the story. At every step, the model generates based on the representation of the given outline, the high-level discourse representation, the preceding story context, and the previous memory. Discourse representation is an encoding of the type of paragraph the current paragraph is, including introduction (_i_), body (_b_), and conclusion (_c_), which is appended to the outline representations at every time step. The preceding story context is the same as the hidden state vectors output by the transformer's attention blocks upon feeding generated sentences into a static GPT-2 model. Finally, the memory is a concatenated vector containing both the generated tokens and an encoded state of the story. In PlotMachines, the conditioning of generation depended on a general outline that includes events and phrases for ease of extraction. Even with the better performance in PlotMachines, the stories can benefit from incorporating a comprehensive plot outline such as the output of an event-based planning system that can improve the generated stories' depth and interestingness.

5.4 Pre-trained Models

The idea of pre-trained language models is to build a "black box" that can be instructed to perform any particular task in that language once it has learned the language. The goal is to build an artificial version of a "well-read" person. First, a substantial amount of unannotated data is fed into the language model (for example, the complete Wikipedia dump). This enables the model to pick up on word usage and general linguistic conventions. The model is now moved to an NLP task, where it is fed a second, smaller dataset tailored to that task. This process is used to fine-tune and produce the final model that can carry out the aforementioned task.
They are superior readers, to put it simply! Using a relatively smaller dataset, a model that trains just on the task-specific dataset must comprehend both the language and the task. On the other hand, the language model already knows the language because it'read' substantial language dumps during pre-training. As a result, the language model can directly adjust itself to match the necessary task and outperforms the SOTA. To enable further processing by machines, each word in NLP must be mathematically represented. By pre-training the models on a different, larger dataset to capture the essence of the language, numerous different techniques have been proposed to produce these embeddings. For instance, Word2Vec embeddings were incredibly popular and were utilised directly for a variety of NLP tasks. These word representations, however, are learned in a broad context and do not convey information relevant to a given activity. This is where a language model's fine-tuning component comes into play. Although using the pre-trained embeddings directly can reduce the size of the entire model, it limits us to only employing generalised word representations. On the other hand, by training on the task-specific dataset, language model fine-tuning enables the user to adjust these word embeddings/representations.

For instance, in a generalised context, the word "current" might have a good relationship with both "news" and "electricity." However, allowing the model to fine-tune the word representations so that "current" and "electricity" match better can help the model perform better for a particular task that discusses electric circuits.

Text generation uses input data to create believable, legible text in a human language. Particularly with the aid of neural generation models based on previously taught language models, the return of deep learning has significantly enhanced this discipline (PLMs). PLM-based text production is thought to be a promising strategy in both academia and business. We conduct a survey on the use of PLMs for text production in this study. We start by outlining three crucial facets of using PLMs for text generation:

- how to preserve input semantics while encoding the input into representations that can be fused into PLMs;

- How to create a PLM that is effective for use as the generation model; and 3) How to efficiently optimise PLMs given the reference text and make sure that the produced texts adhere to specific textual requirements.

Then, we outline the main issues that have arisen in these areas and provide potential remedies. We also provide a list of other helpful sources and examples of PLM-based text creation applications. Finally, we emphasise the areas for future study that will help these PLMs for text production become even better. This thorough overview aims to teach academics interested in text generation issues the fundamental ideas, the key methods, and the most recent advancements in this field based on PLMs.

Neural network models have taken over the dominant text creation approaches with the development of deep learning techniques, and they are incredibly successful at producing natural language, texts. The sequence-to-sequence framework is typically used in deep neural generation models utilising an encoder-decoder system: Prior to generating the target text in the decoder, the encoder maps the input sequence into fixed-sized lowdimensional vectors known as input embeddings on the embeddings of the input. A significant improvement above previous methods is the representation using embeddings statistical methods that make it simpler to deal with potential relationships between inputs and results. So from the above details we can easily guess the power of the pre-trained models which will be shown in the further chapters.

6 Summary

In this paper we describe the in details the exact structure of a movie script. The different parts of a script that is creative in nature and the international format for writing a script. The different methods approached in the task of Automatic Story Generation has been elaborated. The classical Symbolic Approach and the modern Neural Approach. As discussed above the symbolic approaches generally shows lack of creativity and novelty while on the other hand, neural approaches shows lack of coherence and cohesiveness throughout the stories, The neural approaches are found to be rich in creativity and novelty. These are the things are broadly discussed is this report. We have described in details the open websites from where we have downloaded the movie scripts and the process for automatic annotation of movie scripts. The GPT2 and BART model uses the symbolic approaches while the other two uses the Neural approach. As discussed above the symbolic approaches generally shows lack of creativity and novelty while on the other hand, neural approaches shows lack of coherence and cohesiveness throughout the stories, The neural approaches are found to be rich in creativity and novelty. These are the things are broadly discussed is this paper.

References

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Pablo Gervás. 2009. Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–49.

Harry Halpin, Johanna D Moore, and Judy Robertson. 2004. Towards automated story analysis using participatory design. In *Proceedings of the 1st ACM workshop on Story representation, mechanism and context*, pages 75–83.

Carlos León and Pablo Gervás. 2014. Creativity in story generation from the ground up: Non-deterministic simulation driven by narrative. In *ICCC*, pages 201–210.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Joshua McCoy, Mike Treanor, Ben Samuel, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin.

2014. Social story worlds with comme il faut. *IEEE Transactions on Computational intelligence and AI in Games*, 6(2):97–112.

James R Meehan. 1977. Tale-spin, an interactive program that writes stories. In *Ijcai*, volume 77, page 9198.

Janet Horowitz Murray and Janet H Murray. 2017. *Hamlet on the holodeck: The future of narrative in cyberspace*. MIT press.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. *arXiv preprint arXiv:2004.14967*.

Mark Owen Riedl and Vadim Bulitko. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine*, 34(1):67–67.

Judy Robertson and Judith Good. 2003. Using a collaborative virtual role-play environment to foster characterisation in stories. *Journal of Interactive Learning Research*, 14(1):5–29.

Blake Snyder. 2005. *Save the cat*. Michael Wiese Productions Chelsea, Michigan.

Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J Martin, Animesh Mehta, Brent Harrison, and Mark O Riedl. 2019. Controllable neural story plot generation via reward shaping. In *IJCAI*, pages 5982–5988.

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.