The Tomato Festival: Towards using ChatGPT for Long-Form Discourse Generation of Plan-Based Narratives?

Maryam Dueifi[1], Markus Eger[2,*]

[1]Cal Poly Pomona, Department of Computer Science
[2]UC Santa Cruz, Department of Computational Media

Abstract

With Generative AI at the forefront of public conversation, the capability of AI systems to tell stories has seen a surge of interest. OpenAI's ChatGPT provides a user-friendly interface, as well as well-documented API access, and is used widely for generative purposes. In this paper we investigate how well it can actually produce narrative text. We present an approach to take a story plan produced by the Glaive narrative planner and turn it into a novella-length text. We then present a preliminary evaluation of the text output and discuss the challenges and limitations of having it actually be read by humans. Crucially, we show that the text is not comparable to human-written text in terms of grammatical complexity, which we posit to be one possible reason for it not being very enjoyable to read. As part of our work we also encountered several particular challenges that led to misspun tales, which we also discuss in detail.

1. Introduction

Narrative generation has been a topic of interest for AI research for many decades. Meehan's TaleSpin [1] is often cited as the first "story generator", although other approaches have preceded it [2]. Nevertheless, TaleSpin, with its use of character goals and plans has served as the inspiration of a wide variety of plan-based narrative generators. Often, these generators distinguish between the story/fabula part of a narrative, consisting of the events as they happened in the story world, and the *discourse*, i.e. the way that story is actually told, like the text that is produced [3]. More recently, Large Language Models (LLMs) have seen a surge in popularity, including through OpenAI's ChatGPT [4]. These are neural network model architectures termed transformers [5], which can learn correlations between the occurrences of words in a text corpus. This can then be used to answer user queries, by letting the model predict the most likely continuation that follows the question text, producing a response text. However, LLMs can produce all kinds of text, including narrative, when prompted to do so.

While LLMs *can* produce narrative text, the transformer mechanism that they are based on has one significant limitation: When predicting the continuation of a text, the attention head mechanism employed by the model can only look at a limited context preceding the continuation. This context window thus limits how much information the LLM can even see. Essentially, if one tried to generate an entire novel using an LLM with a limited context window, the model would not be able to take the contents of the first few chapters into account when generating the end of the narrative, which may result in events that contradict earlier changes to the world state. Plan-based narratives, on the other hand, use an explicit (logic-based) world model to represent the state of the story world at each time step, and can ensure that only actions that are actually possible to occur are taken by the characters. However, generating the discourse for such a plan-based story often involves templates or other short story fragments, often resulting in repetitive or terse discourse output.

In this paper we present an approach that utilizes OpenAI's ChatGPT to generate the discourse for a story generated by a narrative planner. In particular, we focus on generating long-form narratives, that, at present, reach the length of a novella (about 25 000 words), and follow the story as produced by the planner. Our contribution is threefold: First, we present a novel approach to prompt an LLM using the planner output to expand the narrative to the desired length. Second, we have generated several narratives using our approach and show an evaluation of some of their qualities. Third, and arguably most importantly, during the course of our work we have discovered several limitations of this application of LLMs, and we will discuss them in detail.

2. Related Work

Our work builds on previous work in narrative generation, combining a logic-based story structure generated by a planner, with text generated by an LLM. We will therefore discuss prior work in both of these areas.

2.1. Plan-Based Narrative Generation

When a story is viewed as a (partially ordered) sequence of actions taken by the characters, it has striking similarities to a *plan*, in the AI planning sense: A sequence of actions applied to an initial state to achieve a goal condition [6]. Indeed, there is a large body of work that utilized planners to generate such stories [7]. The main challenge is how to define the "goal" of the story: Often, this is described as a state the story world ought to be in at the conclusion of the story as desired by an author. However, a centralized planner assigning actions to characters to reach this global goal might lead to the characters acting contrary to what common sense would dictate. In a bank heist story, an "efficient" plan might be that the bank teller just delivers the money to the robber's house, but it would likely not make for a compelling story. In TaleSpin, the individual characters have their own goals, preventing them from acting against their own interests [1], but this may not always lead to a story with an actual plot. Another approach, originating with a branch of research by Riedl and Young [8]), instead uses a centralized planner that allows the author to define an overall goal for the story, while also ascribing *intentions* to each character, preventing them from taking actions that do not further their own character-goals. Ware and Young [9] build on this work and also give characters the capability

AIIDE Workshop on Intelligent Narrative Technologies, November 18, 2024, University of Kentucky Lexington, KY, USA
*Corresponding author.
mdueifi@cpp.edu (M. Dueifi); meger@ucsc.edu (M. Eger)

to form plans that they do not end up executing due to some conflict that arises. Other research has also investigated the use of landmarks to guide the generation process [10], incorporating character beliefs [11, 12], and failed actions [13]. Many of these planning-based approaches use the standardized Planning Domain Definition Language, PDDL, [14], and although some use extensions to handle intentions and beliefs, these can be compiled away if needed [15, 16]. For our present work, we use the Glaive narrative planner, which implements intentions and beliefs using an extended PDDL-variant [17], and which comes with a library of standard narrative planning problems, which we will discuss in more detail below.

With a story in hand, the next problem is how to convey this story to an audience, i.e. how to generate the *discourse* [3]. The discourse generation problem actually consists of several parts, including selecting which actions of the story to tell, which to omit, the ordering of the telling in order to convey the necessary information to the audience, and then determining the actual realization of the discourse in the form of text or other media. Research has focused significantly on the first parts, by planning which discourse actions convey the "right" information to the audience [18], how to model suspense [19], flashbacks and flashforwards [20, 21] or focalization [22]. The actual text generation is then typically handled through templates, as was e.g. the case for the CPOCL experiments, or with a simple text planner [23]. Another option is to manually translate the internal representation to text, as was done for TaleSpin [24].

In a more refined proposed model by Barot et al. [25], the authors draw a distinction between the *discourse*, which incorporates the decisions for what to tell, and the *narration*, which is more concerned with *how* to tell the narrative. Our work is best characterized as focusing on the narration, i.e. the surface text realization, based on full story plans. As we will discuss below, this is not a strong limitation for the data set we were working in, but a more sophisticated discourse model could be incorporated in the future. To generate the output text, we make use of Large Language Models, which we will briefly discuss next.

2.2. Large Language Models

Large Language Models (LLMs) are based on a neural network architecture called transformers [5], which add an attention head mechanism to recurrent neural networks, essentially allowing them to learn a probability distribution of words conditioned on the context these words occur in, with varying weights for the context. In recent years, LLMs have seen a surge in popularity, in part due to the availability of OpenAI's ChatGPT, which presents an LLM using a chat-like interface. The inference-capabilities of the LLM are used to predict the most likely continuation to a user query. In practice, if the user enters a question, the most likely continuation, as learned by the model, is an answer to that question, whereas if the user enters instructions, the most likely continuation follows these instructions. While the basic premise is enticing, at present LLMs suffer from a variety of issues, including hallucinations, where they make up people, events, citations, or court cases, and just plain factual inaccuracies. As text-models, they are ill-equipped to reason, or perform calculations. For our purposes, though, these issues are not necessarily problems, as we *want* the model to generate novel content. Indeed, this potential for creativity has been used to generate NPC responses in a murder mystery [26], control game play in interactive RPGs [27], or even to generate entire games using VGDL [28].

3. Our Approach

Our system is able to generate a long-form narrative consisting of multiple chapters. As input we utilize a plan, as obtained by the Glaive narrative planner, as well as an optional genre descriptor. Our discourse generator works in three steps:

1. Convert Glaive plan into chapter descriptions.
2. Generate chapters from descriptions.
3. Summarize and regenerate each chapter.

In each step, our approach utilizes the different roles one can provide when prompting ChatGPT as shown below. To summarize: The *system* role provides the model with high-level guidance, the *assistant* role gives the model context (typically previous model output, but can also be used to demonstrate desirable output to the model), while the *user* role contains the actual prompt the model should respond to.

3.1. Chapter Descriptions

The first challenge when generating discourse from story plans is that the planner output is in the form of a plan, in the case of Glaive this comes in a PDDL-like syntax. The first step is therefore to convert these formal representations into descriptions of a story chapter, where each step in the plan corresponds to one chapter (we will discuss the implications of this below). Rather than having the domain author come up with a mapping of plan actions to a description, we utilize ChatGPT itself to make this mapping. Given a plan step s, e.g. "`(hatch-plan robbie six-shooter brown-horse bank mother-lode)`", we construct the following prompt:

- **System:** You are rephrasing a string of words.
- **User:** Take the following phrase and make it into a coherent sentence: s. Provide only the resulting sentence.
- **Assistant:** In a statement like (accept talia rory village), the meaning is: "Talia accepts Rory's proposal in the village"

The example mapping provides the model with guidance how to interpret. For the example step above, the model produces (variations of) the description "Robbie hatched a plan to rob the bank for the mother lode with his six-shooter and brown horse.". In the next step, we use these steps to generate individual chapters of the story.

3.2. Chapter Generation

Once we have a textual description of each step of the plan, we have the model generate a chapter of the story based on the description. However, when using only the description of the current step, chapters become disconnected, with characters changing frequently between them, needless repetitions, or plain inconsistencies. On the other hand, the context the model can use is also limited, so we cannot provide it with the entire story so far. Instead, we construct the following prompt, incorporating only the text of the immediately preceding chapter:

- **System:** You are a story teller continuing a story.
- **User:** Take the following chapter and make the next chapter, and include dialogue and natural progression: <previous chapter text>.
- **Assistant:** The current chapter is: <i>
- **Assistant:** The current chapter is about: <chapter description>
- **Assistant:** The genre is: <genre description>

For the first chapter, the system prompt is changed to "beginning a story", and no previous chapter text is included. For the last chapter, the model is also explicitly told that it is "ending the story" in the system prompt.

With these first two steps, the system will already produce a discourse following the events of the plan produced by Glaive. However, there will still be noticeable disconnects, as the generation process does not take into account what may follow. Chapters often end with "To be continued...", or even "The end.", even though the narrative has not reached its conclusion. We therefore added another processing step to refine the flow of the narrative.

3.3. Chapter Summarization and Regeneration

One key limitation with LLMs for our work is their limited context. However, even with larger context size, providing the model with more input does not guarantee more desirable output. Nevertheless, in order to improve the consistency of the narrative, we take each story chapter and have the model rewrite it in context. For each chapter, we first ask the model to provide a summary of the events that happen in it, with the simple prompt "Create a short summary of the following chapter: <chapter text>". We then ask the model to rewrite each existing chapter with the following prompt:

- **System:** You are a story teller remaking chapters.
- **User:** Rewrite the following chapter: <chapter text>
- **Assistant:** Keep in mind that the <relative position> chapter is: <summary of relevant chapter>

Where the assistant-phrase is present up to four times: We take the context of the current chapter to be the two immediately preceding and two immediately following chapters, and include their summaries in the prompt. This causes the model to take the events of these chapters into account when rewriting the current one.

4. Results and Discussion

Depending on the input plan, our approach is able to produce a narrative telling that may reach the length of a short novella. This makes evaluating the output challenging, as it requires reading through hundreds of pages of text and determine its quality. Before we go into more detail about these challenges, and the evaluation we performed, we will first discuss the input data we used and then show some sample output to demonstrate what our approach is capable of. However, we would be remiss to not also discuss limitations and challenges that remain, which we will do to conclude this section.

4.1. Input

We use the Glaive narrative planner to produce story plans for use with our system [17]. Glaive comes with 7 standard narrative planning problems, of which we use four for our experiments:

- **Fantasy**: A story world set in a magical kingdom, with two lovers, Talia and Rory, and a monster, Gargax, guarding a treasure.
- **Heist**: A story world set in the American old west, set in a town with a bank, a saloon, and options for the characters to rob the bank, cheat at poker, steal valuables and exchange them for money, and for the sheriff to arrest criminals.
- **Raiders of the Lost Ark**: A story world based on the movie "Indiane Jones: Raiders of the Lost Ark", set in 1936, with a powerful artifact, the Ark of the Covenant, being chased after by Indiana Jones (at the behest of the US Army), and the Nazis.
- **Western**: Another story world set in the American old west, featuring snakes that can bite characters, and anti-venom that must be obtained to heal the snake bite.
- (Not Used) **Aladdin**: A story world based on the tale "Aladdin" from 1001 Nights, with a king, a woman called Jasmine, a genie that can fulfill wishes, and a hero character. We did not use this domain because "Jasmine" and "Genie" were specific enough for ChatGPT to recall the story from its training set. We will discuss why this did not consistently happen for Raiders of the Lost Ark below.
- (Not Used) **Best Laid Plans**: A story world consisting of a goblin minion that must obtain hair tonic for their warlock overlord. We did not use this domain, as the vast majority of actions performed in valid plans are move-actions, which did not lead to very interesting narratives.
- (Not Used) **Space**: A domain ostensible set in space with volatile planets and aliens. However, this domain is underdeveloped, with the sample problem leading to a solution of only two steps, far shy of an interesting narrative, and we therefore did not use this domain either.

While we did not use the three last domains for our main experiments, we still attempted to use them, and we will elaborate more on some of the problems we encountered below.

4.2. Results

Based on the narrative domains discussed above, we generated 10 text outputs for every one of the four domains, using the solution plan included with Glaive (it is, of course, possible, to generate more/different solution plans, to produce more varied stories). For our purposes, we were interested in how well the translation of plan steps to chapters would work, what the chapters produced by ChatGPT would look like, and how consistent they are across the narrative. Due to the recent release of ChatGPT 4o we performed our experiments twice, once using ChatGPT 3.5, which has much faster processing times, and once with ChatGPT 4o, to make use of the newest version of the model.

First, despite only being given a single example step, the model is able to translate plan steps into chapter descriptions

Table 1
Translation of the first 10 plan steps of the heist story into chapter descriptions as generated by ChatGPT 3.5 and 4o. In most cases, the descriptions were identical; rows marked with * indicate where the 4o version of ChatGPT produced significantly different output, which is listed there.

Plan step	Chapter description
(hatch-plan robbie six-shooter brown-horse bank mother-lode)	Robbie hatched a plan to rob the bank with his trusty six-shooter and brown horse to strike the mother lode.
(open jill general-store)	Jill opened the general store.
(move-once sally main-street bank)	Sally moved once on Main Street near the bank.
(withdraw-money sally bank dress-money)	Sally withdrew money from the bank to pay for her dress.
*(withdraw-money sally bank dress-money)	Sally withdrew money from the bank in her dress.
(move-once sally bank main-street)	Sally moves once on Main Street.
(sell sally anne tomatoes tomato-money main-street)	Sally Anne sells tomatoes for money on Main Street.
(move-once robbie main-street dark-alley)	Robbie moved once from Main Street to a dark alley.
(hide-in-dark-alley robbie dark-alley)	Robbie hides in a dark alley.
(move-once barney barneys-room saloon)	Barney moves once in Barney's room at the saloon.
*(move-once barney barneys-room saloon)	Barney moved once from Barney's room to the saloon.
(pickpocket robbie sally dress-money main-street dark-alley)	Robbie and Sally pickpocket dress money on Main Street in a dark alley.
*(pickpocket robbie sally dress-money main-street dark-alley)	Pickpocket Robbie stole Sally's money from her dress in a dark alley on Main Street.

very well, as shown in Table 1. In most instances, the descriptions produced by the two model versions were almost identical, with only three steps being described differently. Notably, the models sometime struggle with distinguishing subject and object of an action, but the resulting descriptions are still serviceable to turn them into chapters.

The second step is to take the chapter descriptions and having the model produce narrative text that expands the description into an entire chapter. At first glance, this produces output that reads like an actual story. For example, one of our produced texts for the heist story starts as follows (as told by ChatGPT 3.5):

> In the dusty town of Redwood Creek, the sun beat down mercilessly on the wooden buildings that lined the main street. Robbie stood outside the saloon, wiping the sweat from his brow as he watched the townsfolk go about their business. He had been planning this heist for months, and today was the day he would finally put his plan into action.
>
> "Hey Robbie, you lookin' mighty serious there," a voice called out from behind him. Robbie turned to see his old friend, Jake, approaching with a grin on his face.
>
> "Hey Jake, just thinking about our little job today," Robbie replied, trying to keep his voice casual.
>
> Jake's eyes lit up with excitement. "You mean the bank job? I thought you were just jokin' about that."
>
> Robbie shook his head. "No joke, Jake. We're gonna hit that bank and make off with enough gold to set us up for life."
>
> Jake's grin widened. "Count me in, partner. When do we make our move?"
>
> Robbie glanced around to make sure no one was listening in. "Tonight, after the sun goes down. We'll ride in, guns blazing, and take what's rightfully ours."

The same story, as told by ChatGPT 4o starts like this:

> The sun dipped low over the dusty town of Red Rock, casting long shadows across the wooden storefronts and dirt-packed streets. The air was thick with the scent of sagebrush and the distant promise of rain that never seemed to come. Robbie McGraw sat on the creaky porch of the Rusty Spur Saloon, his six-shooter resting comfortably in its holster and his brown horse, Whiskey, tethered to the hitching post nearby.

Of particular note is that while the input only mentioned a character "Robbie" by name, both versions of the model fill in additional names for the town, other characters, or even the horse, as needed. Overall, for the heist story in particular, the models produce text that is very compelling in parts, but falls short of actually being interesting, as we will discuss in the next section.

4.3. Evaluation

One key point of our approach is that while we generate chapters one by one, we then have the model rewrite them in the context of the surrounding chapters. The purpose of this is to ensure a higher level of consistency. Even though the model is given the previous chapter when writing the next one, characters are not used continuously, and instead new characters are introduced, or the role of a character is changed between chapters. The rewrite attempts to address this issue by putting the chapter in context of what happens before *and* after it. One way to show this effect is to determine how often individual characters show up, and how the rewrite affects the output. Table 2 shows some basic statistics of the generated narratives. We used spaCy[1] to perform named entity recognition for each chapter, and tracked the different character's occurrences across chapters. We call characters that appear in more than 30% of chapters "main characters", as plans often include actions performed by other characters, and characters that only show up in one or two chapters "incidental" characters. The

[1] https://spacy.io/

Table 2
Some basic statistics of the generated narratives, averaged over 10 outputs for each narrative: Number of words, the number of main (appear in more than 30% of chapters), incidental (appear in only one or two chapters) and other characters, for the initial version of the narrative as well as the regenerated one (indicated with RW).

Narrative	ChatGPT	Words	Main	Other	Inc.	Words (RW)	Main (RW)	Other (RW)	Inc. (RW)
Ark	3.5	4051.9	3.6	1	4.9	3303.3	3.2	1	4.3
Ark	4o	7646.7	4.8	1	6.4	7006.4	4.9	1	6.2
Fantasy	3.5	3193.4	3.3	0.9	1.3	2612.2	4.1	0.9	0.8
Fantasy	4o	5619.6	4.7	1	5.4	5085.4	6.3	1	3.6
Heist	3.5	15601.1	3.3	5.7	10.9	13060.8	3.2	5.5	12.7
Heist	4o	27526.5	6.9	5.2	15.5	27456.5	6.8	5.1	15.2
Western	3.5	4219.2	4.5	1	2.7	3628.4	4.4	1	2.8
Western	4o	6840.5	6.3	1	5.1	6507.4	6.1	1	5.8

table shows several effects of the rewrite, as well as some differences between the two model versions: In every instance, the rewritten narrative is shorter than the original, as the model is able to remove some redundancy. Characters are also utilized slightly differently, as shown by the changing number of main and incidental characters, but a more detailed analysis of this effect is still an open problem. Also noteworthy is that the newer model is significantly more loquacious than the older version, producing almost double the output given the same input story steps.

As our approach is able to generate narratives consisting of thousands of words, a more detailed evaluation is very challenging. Perhaps the most desirable form of evaluating the produced narratives would be by gathering feedback from human readers, but we encountered two obstacles to this: First, the sheer quantity of text the readers would have to go through is beyond any reasonable compensation we could offer, particularly, because second, upon closer inspection, the writing is not very good. Rather than subject volunteer participants to what we do not believe to be good literature, we set out to quantify *why* the writing does not seem to be enjoyable. Below we will detail some perhaps more anecdotal evidence, but we also attempted to quantify some properties of the writing itself. Subjectively, the rhythm of the writing seems artificial (which it is), and we believe this is due to the repetitive sentence structure. The Frazier score is a measure for syntactical complexity [29], and measures, broadly speaking, the depth of the parse tree of a sentence. Higher scores therefore indicate higher grammatical complexity, while simpler sentence structures are scored lower. We used nltk[30] and Stanford CoreNLP [31] to compute the parse tree, and computed the Frazier score for each sentence of the generated narratives. For comparison purposes, we also computed the Frazier scores for books written by human authors: Mary Shelley's *Frankenstein* [32]), Jane Austen's *Pride and Prejudice* [33]), Victor Hugo's *Les Miserables* (English translation by Isabel Florence Hapgood) [34] and Sir Arthur Conan Doyle's *The Adventures of Sherlock Holmes* [35]. We believe that these books are a good representation of non-trivial literature, as they are considered classics and literary achievements, yet still readable by a dedicated reader. While the style and length of the books may vary, the average Frazier score of all four books fell between 7 and 8. For comparison, the average Frazier score of the narratives generated by ChatGPT 3.5 fell between 14 and 18 for the different narratives, and while ChatGPT 4o did produce less convoluted sentences, its average Frazier score still ranged between 10 and 12. Figure 1 shows the distribution of Frazier scores across each individual narrative, together with the mean and 95% confidence intervals. It can be seen that human authors tend to use a rather even mix of more and less complex sentences, while the models tend to eschew simpler constructs in favor of sentences consisting of more nested clauses. Note that neither higher nor lower Frazier scores are inherently "better", and this evaluation only serves to provide a comparison with some samples of "typical" (good) human writing. In our main experiments we avoided instructing the model to imitate human authors due to some ethical concerns, but for comparison reasons we added the instruction to write in the style of each of the authors of the four books, and performed the same analysis. Based purely on grammatical complexity, the model does not seem to capture the same writing style, and instead further increases sentence nesting.

4.4. Misspun Tales

While we believe that the results we show above already constitute a novel contribution, various problem cases we uncovered may also be of interest to future researchers. First, while our approach often results in narratives that follow the given structure, this is not always the case. Since the model is given the previous chapter input, as well as the desired next step, it has to perform a trade-off in how much attention to give to each, and at times the "most likely" continuation ignores the plan steps entirely. In one particularly noteworthy instance, ChatGPT 4o took the Heist narrative, had the bank robbery take place in chapter 3, followed by an escape by sea onto a pirate ship, which then turned into a fantasy story to chase after a powerful artifact, concluding with (as summarized by the model):

> In Chapter Thirty: The Convergence, the town of Port Meridian buzzes with anticipation as the day of the Great Unveiling approaches. Robbie and Talia, along with the committee, work tirelessly to decipher the Heart of the Ancients' inscriptions. They uncover a crucial passage about the "Convergence of Realms," a moment when the boundaries between their world and the Ancients' will blur, unlocking immense knowledge but also posing significant dangers. As the committee intensifies their efforts, Robbie and Talia resolve to ensure that this newfound wisdom is used ethically and for the greater good, heralding a new era of unity and potential.

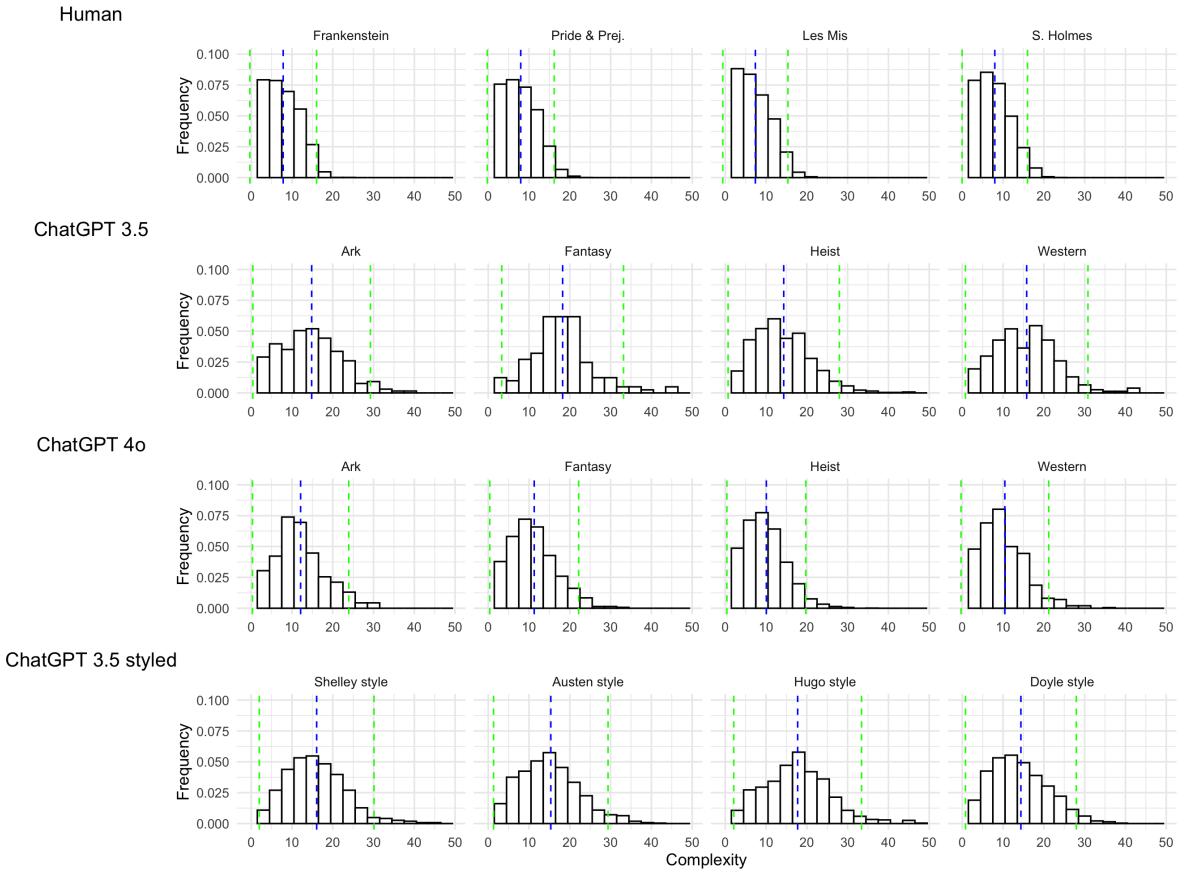

Figure 1: Distribution of grammatical complexity across sentences with mean values and 95% confidence interval, as measured by the Frazier score. In the first row, we provide distributions of classical, human-written books as comparison. The second row shows the distribution in narratives produced by ChatGPT 3.5, while the third row shows the same for narratives produced by ChatGPT 4o. For the fourth row, we additionally instructed the model (ChatGPT 3.5) to write the heist narrative in the style of the author of each of the books from row one. Neither higher nor lower complexity scores are inherently "better", but the distribution of more and less complex sentences is a property of an author's writing.

A minor point that this conclusion also demonstrates is the tendency of the model to lead to happy endings, that reassure the reader that whatever power or treasure is obtained will be used ethically. While not "wrong", the persistent mention of this is not well-placed in all story contexts. We suspect that this is due to some of the "safeguards" OpenAI has integrated into their system, to prevent (some) unethical output. We do not disagree with this choice, but it also shows that controlling LLM output to make it suitable for all applications is challenging.

Generally, control is a major issue when using an LLM. Our approach to rewrite chapters to make characters behave more consistently is not the only thing one could do. We attempted to tell the model outright which characters exist and what their roles are. However, this led to two problems: First, the characters have to come from somewhere. If the domain author is tasked with providing a character list, they will need to foresee which larger cast of the characters the model might need, which may also lead to higher repetitiveness of the generated stories. Our solution to that was to let the model generate an (ideally) varied list of characters to use. However, the model does favor certain names for different genres (Robbie's partner in the heist story is usually called "Hank" or "Jake", despite neither showing up in the input). The second problem, though, is that the model does not seem to have enough context to work with any character list that is given to it. When instructed to use specific names, the text might initially use the provided names, but they often change back to the names favored by the model for each particular narrative.

Even when the model follows the provided trajectory and uses characters consistently, though, it struggles with keeping a cohesive tone. Step 7 in the heist plan is "(sell sally anne tomatoes tomato-money main-street)", which consists of a single transaction. In one instance, the model took this idea and just "ran with it", turning one character selling tomatoes into the three protagonists hosting a tomato festival where they sell produce together with the local farmers in order to finance their travels:

> With renewed purpose, the trio embarked on a mission to gather more tomatoes and secure a venue for the tomato festival. They approached local farmers, explaining their plan and offering a fair share of the profits. To their surprise, the farmers were intrigued by the idea and agreed to contribute their tomato harvest.

By itself, the tomato festival is a reasonable interpretation of the given story step, but it stood out in context, as the

preceding chapter is titled “The Enigmatic Stranger”, and the following chapter “The Mysterious Stranger”. Overall, the model favored a darker, grittier narrative, which made the tomato festival seem even more out of place. On the other hand, as these other two chapter titles already indicate, encounters with strangers, risk, or danger are all narrative devices the model employs frequently. The narrative in question contains three consecutive chapters titled “A Risky Proposition”, which is then followed by “The Perilous Journey”.

It is already known that ChatGPT often produces inaccurate information [36], but this also raises issues even when it produces fiction. It may, for example, suggest ordering whiskey to sober up:

> Robbie led Barney to a table in the corner, away from prying eyes. He signaled the bartender for two whiskeys, hoping the strong drink would sober Barney up enough to have a coherent conversation.

On the flip side, as some of the examples are based on existing media, which OpenAI may have included in the data they used to train the model, the resulting narrative may just reproduce this existing data, rather than producing a new one. This was particularly challenging with the Aladdin story, but for the Indiana Jones-based story another interesting phenomenon occurred: Rather than interpreting “Indiana” as a name (even in context with the Ark and the location of Tanis), ChatGPT would take it as the US state of Indiana, and then either name a town there “Tanis” or turn “Tanis” into a character, as in this example:

> The sun was just beginning to rise over the horizon, casting a golden hue across the small town of Maplewood, Indiana. The streets were still quiet, with only the occasional chirping of birds breaking the silence. Tanis stood at the edge of her driveway, her backpack slung over one shoulder and a map of Indiana clutched in her hand. She took a deep breath, feeling the crisp morning air fill her lungs. Today was the day she had been waiting for.
>
> “Are you sure about this, Tanis?” Her best friend, Mia, asked as she approached. Mia’s eyes were filled with concern, but there was also a hint of excitement. “It’s a long way to Indianapolis, and you know how unpredictable things can get.”

Overall, using ChatGPT for narrative generation seems to produce reasonable output on the surface, but once one looks at the text more closely, problems keep arising almost fractally, as one digs deeper on a problem another one shows up. The model has a general notion of what a “good” narrative would look like, but no understanding of flow, composition, coherence, common sense, or purpose. Attempts to rectify this by better prompts are only partially successful, as providing too much instruction to the model makes it more likely to ignore parts of it, while providing too little guidance results in rambling. We will conclude with the entirety of chapter 1 for one iteration of the Ark story (using ChatGPT 4o):

> The genre is action, adventure, and mystery. (sic!)

This happened exactly twice in all of our experiments, all other attempts produced actual chapter text.

5. Conclusion and Future Work

We have presented an approach to using ChatGPT, to produce long-form text for discourse generation – or, more precisely, surface text realization. Our approach takes story steps produced by a narrative planner and tasks the model with first translating the abstract step output to descriptions of individual chapters, and then turn these descriptions into actual chapter text. We perform another pass over the chapters where we ask the model to summarize each chapter, and then rewrite it using the summaries of the preceding and following chapters as additional context. We show several example outputs of our model, and discuss how challenges with an evaluation of rather long texts that are not particularly well written. Crucially, we also investigate why the texts do not appear to read well, and show an analysis of the grammatical complexity of the generated narratives, which tend to be more complex than comparable human-written literature. Finally, we discuss a myriad of other problems we encountered that led to narratives that were ill-formed, illogical or incongruous.

While our work presents a somewhat bleak outlook on the current state of narrative generation using LLMs, we believe these insights are crucial to understanding what makes text seem artificial. The alien-ness of AI text may have been intuitively understood, but our work attempts to quantify it, which may in turn lead to improvements in output, or - perhaps more importantly - highlight the importance of human authorship. Our approach also focused on surface text realization without taking larger questions of discourse generation into account. Some repetitiveness may also be alleviated by not having the model generate an entire chapter for rather trivial move actions.

References

[1] J. R. Meehan, TALE-SPIN, an interactive program that writes stories., in: IJCAI, volume 77, 1977, pp. 91–98.

[2] J. Ryan, Grimes’ fairy tales: a 1960s story generator, in: Interactive Storytelling: 10th International Conference on Interactive Digital Storytelling, ICIDS 2017 Funchal, Madeira, Portugal, November 14–17, 2017, Proceedings 10, Springer, 2017, pp. 89–103.

[3] R. M. Young, Story and discourse: A bipartite model of narrative generation in virtual worlds, Interaction Studies 8 (2007) 177–208.

[4] OpenAI, Gpt-4 technical report, 2024. arXiv:2303.08774.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[6] M. Lebowitz, Story-telling as planning and learning, Poetics 14 (1985) 483–502.

[7] R. M. Young, S. G. Ware, B. A. Cassell, J. Robertson, Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives, Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative 37 (2013) 41–64.

[8] M. O. Riedl, R. M. Young, Narrative planning: Balancing plot and character, Journal of Artificial Intelligence Research 39 (2010) 217–268.

[9] S. Ware, R. Young, CPOCL: A narrative planner supporting conflict, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 7, 2011, pp. 97–102.

[10] J. Porteous, M. Cavazza, Controlling narrative generation with planning trajectories: the role of constraints, in: Interactive Storytelling: Second Joint International Conference on Interactive Digital Storytelling, ICIDS 2009, Guimarães, Portugal, December 9-11, 2009. Proceedings 2, Springer, 2009, pp. 234–245.

[11] S. G. Ware, C. Siler, The sabre narrative planner: multi-agent coordination with intentions and beliefs, in: AAMAS Conference proceedings, 2021.

[12] H. Mohr, M. Eger, C. Martens, Eliminating the impossible: A procedurally generated murder mystery., in: AIIDE Workshops, 2018.

[13] R. Sanghrajka, R. M. Young, B. Thorne, Headspace: incorporating action failure and character beliefs into narrative planning, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 18, 2022, pp. 171–178.

[14] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. Sri, A. Barrett, D. Christianson, et al., Pddl| the planning domain definition language, Technical Report, Tech. Rep. (1998).

[15] P. Haslum, Narrative planning: Compilations to classical planning, Journal of Artificial Intelligence Research 44 (2012) 383–395.

[16] M. Christensen, J. Nelson, R. Cardona-Rivera, Using domain compilation to add belief to narrative planners, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 16, 2020, pp. 38–44.

[17] S. Ware, R. M. Young, Glaive: a state-space narrative planner supporting intentionality and conflict, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 10, 2014, pp. 80–86.

[18] J. Niehaus, R. M. Young, A method for generating narrative discourse to prompt inferences, in: Proceedings of the Intelligent Narrative Technologies III Workshop, 2010, pp. 1–8.

[19] Y.-G. Cheong, R. M. Young, A computational model of narrative generation for suspense., in: AAAI, 2006, pp. 1906–1907.

[20] B.-C. Bae, R. M. Young, A computational model of narrative generation for surprise arousal, IEEE Transactions on Computational Intelligence and AI in Games 6 (2013) 131–143.

[21] H.-Y. Wu, M. Young, M. Christie, A cognitive-based model of flashbacks for computational narratives, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 12, 2016, pp. 239–245.

[22] M. Eger, C. Barot, R. Young, Merits of a temporal modal logic for narrative discourse generation, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 11, 2015, pp. 23–29.

[23] D. K. Elson, Modeling narrative discourse, Columbia University, 2012.

[24] N. Wardrip-Fruin, Reading digital literature: Surface, data, interaction, and expressive processing, A companion to digital literary studies (2013) 161–182.

[25] C. Barot, C. Potts, R. M. Young, A tripartite plan-based model of narrative for narrative discourse generation, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 11, 2015, pp. 2–8.

[26] S. R. Cox, W. T. Ooi, Conversational interactions with npcs in llm-driven gaming: Guidelines from a content analysis of player feedback, in: International Workshop on Chatbot Research and Design, Springer, 2023, pp. 167–184.

[27] X. Peng, J. Quaye, W. Xu, C. Brockett, B. Dolan, N. Jojic, G. DesGarennes, K. Lobb, M. Xu, J. Leandro, et al., Player-driven emergence in llm-driven game narrative, arXiv preprint arXiv:2404.17027 (2024).

[28] C. Hu, Y. Zhao, J. Liu, Generating games via llms: An investigation with video game description language, arXiv preprint arXiv:2404.08706 (2024).

[29] L. Frazier, Syntactic complexity, Natural language parsing: Psychological, computational, and theoretical perspectives (1985) 129–189.

[30] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.

[31] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

[32] M. Shelley, Frankenstein; Or, The Modern Prometheus, Penguin, 1818.

[33] J. Austen, Pride and Prejudice, T. Egerton, Whitehall, 1813.

[34] V. Hugo, Les misérables, Thomas Y. Crowell & Co., 1887. Translation by Isabel Florence Hapgood.

[35] A. C. Doyle, The Adventures of Sherlock Holmes, George Newnes, 1892.

[36] M. T. Hicks, J. Humphries, J. Slater, Chatgpt is bullshit, Ethics and Information Technology 26 (2024) 38.