Large Language Models are Incoherent Storytellers

Nisha Simon[†,*]
[†] School of Computing, Queen's University, Canada

Abstract

At first glance, it would seem that modern LLMs can assiduously parrot back output based on data that has previously been given to them as inputs. However they are still a long way away from coherently generating long form narratives that match defined parameters, unless they are first given a certain amount of guidance. By combining Automated Planning with Natural Language Text Generation by LLMs we can create logical, believable, and coherent stories that can be used in a wide variety of domains, for a large range of applications.

Keywords: Large Language Models, Automated Storytelling, Automated Text Generation, Automated Planning, Interactive Narratives.

1. Research Problem

Research Problem: LLMs cannot maintain coherence over longer narratives while staying within specified parameters. They are prone to repetition, stilted language, bias, and the production of toxic or offensive outputs [1, 2]. The significance of our research problem is that by combining Automated Planning and Natural Language Text Generation, we can provide a scaffolding to guide the LLM to produce pertinent, logical, coherent, and believable outputs or narratives.

Background: Automated Planning problems are represented using the Planning Domain Definition Language (PDDL) [3]. Planning problems use two files written in PDDL format: the *Domain* file and the *Problem* file. The Domain file contains the *requirements, types, predicates* and *actions*, while the Problem file holds the *objects*, the *initial state* and the *goal*. A particular domain could have multiple problems associated with it. The domain and problem files are fed into an automated planner, and the planner then produces a *plan* (typically represented by a sequence of actions or steps) that lead to the goal state.

A classical planning problem P is represented as a tuple denoted by $\langle F, A, I, G\rangle$. **F** is the *set of fluents* or items that can be either TRUE or FALSE in the domain. **A** is the *set of actions* or what the agent is allowed to do in the given environment. **I** is the initial state. **G** is the goal the agent is trying to achieve or the *set of fluents* that must be TRUE at the end of the planning process. An action **a** in the set of actions **A** has three characteristics: PRE(a): the preconditions of action **a** or the set of fluents that must hold to execute action **a**, DEL(a): the set of fluents that are removed from the current state when action **a** is executed, or the '*delete effects*' of action **a**, and ADD(a): the set of fluents that are added to the current state when action **a** is executed, or the '*add effects*' of action **a**. If PRE(a) $\subseteq$ s, the agent can take action **a**. We *progress* from a state ***s*** to state s' using action **a** by removing every fluent that **a** deletes, and then adding every fluent that **a** adds. That is to say *Progress* (s,a) = (s $\setminus$ DEL(a)) $\cup$ ADD(a). The goal is achieved when G $\subseteq$ s. In Fully Observable Non-Deterministic Planning (FOND), we extend classical planning to allow actions to have more than one outcome, thus potentially leading to more than one successor state at execution time. The keyword '*oneof*' indicates multiple possible effects. By including non-determinism, the plan now takes the form of a decision tree instead of a sequence of actions [3].

[*] nisha.simon@queensu.ca

2. Proposed Solution and Approach

We build a model that provides inputs to the Large Language Model that are more context-dependent, and that incorporate commonsense knowledge. Automated Planning can be applied to Natural Language text generation in order to create believable and coherent narratives (stories) based on the given application. Our solution is novel because we use various types of planning methodologies such as Fully Observable Deterministic (FOD) Planning and Fully Observable Non-Deterministic (FOND) Planning, to produce logical, believable and coherent stories. The details of the system architecture and the step by step methodology are shown in Figure 1.

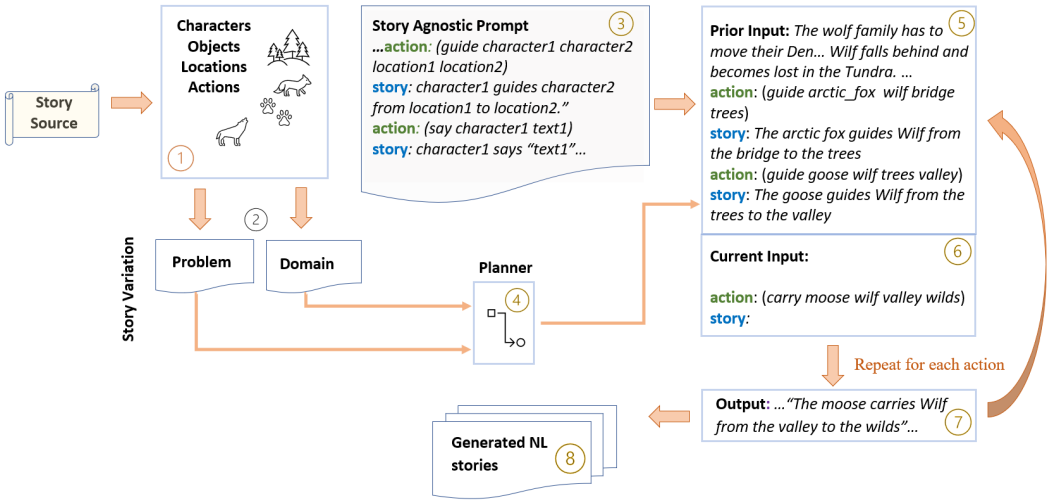

Figure 1. System architecture showing the various components that are used to generate the natural language story. From the original story source, Step 1: characters, objects, locations, and actions are manually extracted. Step 2: *Problem* (story variation) and *Domain* (story mechanics) files are manually created. Step 3: Story-agnostic prompts are created by hand as the initial input to the LLM in order to provide background and style information. Step 4: The planner automatically creates a valid plan. The output of the planner for the FOD Stories are represented by the 'action' lines above. Step 5: prior inputs, along with story agnostic prompts, and Step 6: the actions of the plan (Current input), which is the current action that is being processed, are iteratively used as input prompts into the LLM, to Step 7: generate a natural language story. The resulting output sentences are then collected together and compiled into a plain text file in Step 8 to form the complete generated story.

We began with a selection of children's stories such as '*The Way Home for Wolf*' [4], '*Robin Hood and the Golden Arrow*' [5], and '*The Paper Bag Princess*' [6] since these included simple, basic vocabulary and grammatical constructs. For the initial FOD stories, the planner used was the online solver[1] [7] and the LLM models used were GPT-J-6B eleuther [8] through a browser interface and BLOOM from Hugging Face [9]. For the FOND stories, an automated FOND planner (in this case, an extension to the *PRP* planner) [10] was used along with an LLM (specifically *gpt2-x1* [11]) via the *Hugging Face API* [2]. The PRP planner extension is an off-the-shelf, state-of-the-art planner. It should be noted that the particular LLM and planner that are used are not a key contribution of the solution, as both these components can be switched out at will and can be considered a 'black box' component of our system.

[1] https://solver.planning.domains/
[2] https://huggingface.co/models

3. Progress and Preliminary Results

Our work to date includes stories that were created with both FOD planning [12] as well as with FOND planning. We used both quantitative and qualitative metrics to evaluate our system. The evaluation metrics used include part-of-speech tags of nouns and verbs to indicate how many of the characters, objects, locations (nouns), and actions (verbs) from the plan are reflected in the output of the LLM, i.e. we evaluate how well the generated story from the LLM mirrors the output of the planner. Greater duplication represents greater accuracy of the LLM in reflecting the planner's output. The generated stories are also judged on whether or not they achieved their required author and character goals, as well as on coherence. Castricato, Frazier, Balloch, Tarakad, and Riedl define coherence as '*any perceivable relationship between events in a story*' [2]. The output of the LLM shows that when its inputs are provided as actions from generated plans, both the author and character goals are achieved.

Story	POS tag	Plan	LLM story
The Way Home for Wolf	Noun	18	18
	Verb	2	3
Robin Hood and the Golden Arrow	Noun	8	7
	Verb	6	7
Paper Bag Princess	Noun	6	7
	Verb	10	19

Table 1. Number of nouns and verbs found in the PDDL plan that are successfully duplicated in the LLM output.

We observe from Table 1 that the guided LLM output captures almost all the nouns such as *Goose, Musk Ox* etc., and verbs e.g. '*carry*' and '*guide*' from the PDDL plan. The generated natural language stories also display the required qualities of coherence and consistency. The output was considered to be 'coherent' if there were no grammatical or logic errors e.g., verbs and nouns were in agreement, and a character could not be in two locations at the same time. Our generated outputs, such as the sample in Listing 1, stand in contrast to the output where the LLM is allowed free rein and therefore devolves into repetition. In the absence of relevant PDDL input prompts, the LLM loses the story thread and instead resorts to basic repetition with no regard for the plot outline or for the prior actions that have already occurred in the story. It should be noted that the text of the LLM output should be considered the 'average' or representative output, since the generated output of the LLM may vary slightly every time the process is repeated, especially for more complex stories like '*Robin Hood and the Golden Arrow*' and '*The Paper Bag Princess*. Although the gist and the main thread of the story are maintained over several iterations, exactly the same wording may not be generated each time.

Listing 1: Example of a natural language story that has been generated by the LLM based on input actions from a valid plan for the '*Robin Hood and the Golden Arrow*' story

```
The sheriff announces a archery contest.
Robin Hood learns about the archery contest.
Robin Hood disguises himself as a beggar to enter the archery contest.
Robin Hood participates in the archery contest
The arrow hits the target at the center.
Robin Hood wins the golden arrow.
```

Beyond the quantitative metrics explored above, we also conducted a human evaluation of the initial generated stories based on their coherence and believability. We based our

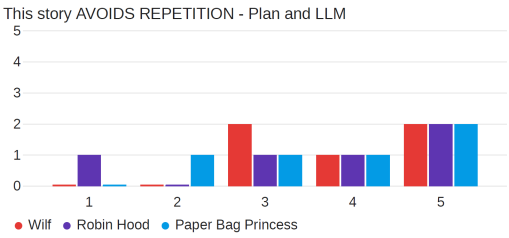

Figure 2. Results for the survey question *"This story avoids repetition"*

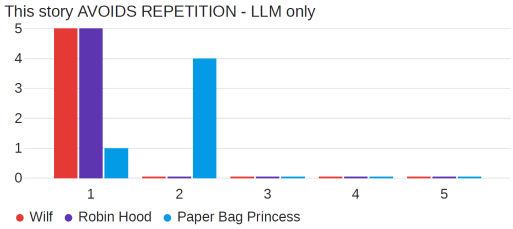

Figure 3. Results for the survey question *"This story avoids repetition"*

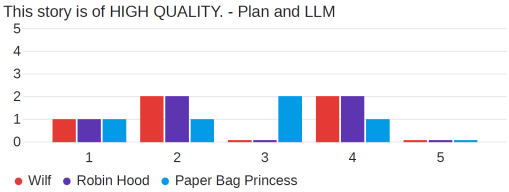

Figure 4. Results for the survey question *"This story is of High Quality"*

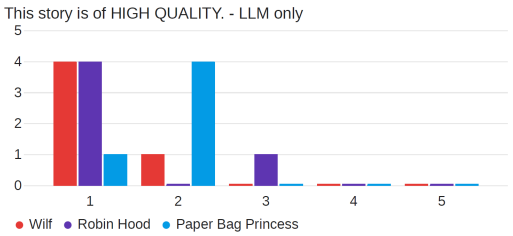

Figure 5. Results for the survey question *"This story is of High Quality"*

exploration on the method that was followed by Purdy, Wang, He, and Riedl (2018) by asking evaluators to respond to statements about the stories. A score of "5" indicates complete agreement with the statement. "*The first three questions correspond to grammaticality, temporal ordering, and local contextuality, respectively*" [13]. Participants were selected from a pool of university students in a Computing program. Interested participants were given an online questionnaire that asked them to rate generated stories. Stories that were generated solely by the LLM's response to an input prompt were compared with stories where the prompts to the LLM were based on the output of a valid plan, and the participants were unaware of which story came from which source (or, indeed, what was generating the stories at all). We found that the human evaluators graded the stories that were generated using a Plan and an LLM higher in general on a variety of metrics than those that were generated ony with an LLM. A sample of the results from our questionnaire is shown in Figures 2, and 3 where the evaluators judged the stories created with both a valid plan and an LLM to have less repetition than stories that were created with only an LLM, and also in Figures 4 and 5, where the evaluators judged the stories created with both a valid plan and an LLM to be of higher quality than stories that were created with only an LLM.

Our current exploration involves FOND planning and Choose Your Own Adventure (CYOA) stories [14] in the form of an interactive text-based game where the user takes on a fictional persona and attempts to make the correct choices to move from a starting state to the 'end game' or goal state. Our initial proof of concept results indicate that providing a valid plan to the LLM allows it to generate logical options that allows the user to successfully navigate the game from start to end within the constraints of the given environment.

Future Work: This research opens the door to many planning-oriented extensions. Future work will include Epistemic Planning giving more believable agent understanding, including the goals of deception/misconception in epistemic planning to tell better stories, and the inclusion of Linear Temporal Logic constraints to direct how a story might unfold.

References

[1] A. Olmo, S. Sreedharan, and S. Kambhampati. "GPT3-to-plan: Extracting plans from text using GPT-3". In: *arXiv preprint arXiv:2106.07131* (2021).

[2] L. Castricato, S. Frazier, J. Balloch, N. Tarakad, and M. Riedl. "Automated Story Generation as Question-Answering". In: *arXiv preprint arXiv:2112.03808v1* (2021).

[3] P. Haslum, N. Lipovetzky, D. Magazzeni, and C. Muise. *An Introduction to the Planning Domain Definition Language*. Morgan & Claypool, 2019. ISBN: 9781627058759. URL: http://www.morganclaypoolpublishers.com/catalog_Orig/product_info.php?products_id=1384.

[4] R. Bright and J. Field. *The Way Home for Wolf*. Vol. 1. Scholastic Press, 2020.

[5] R. D. San Souci and E. Lewis. *Robin Hood and the Golden Arrow*. Vol. 1. Scholastic Press, 2010.

[6] R. Munsch and M. Martchenko. *The Paper Bag Princess*. Vol. 1. Annick Press, 1980.

[7] C. Muise. "Planning.Domains". In: *The 26th International Conference on Automated Planning and Scheduling - Demonstrations*. 2016.

[8] B. Wang. *Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX*. https://github.com/kingoflolz/mesh-transformer-jax. May 2021.

[9] M. e. a. Mitchell. *BigScience Language Open-science Open-access Multilingual (BLOOM) Language Model*. 2022. URL: https://huggingface.co/bigscience/bloom.

[10] C. Muise, S. McIlraith, and C. Beck. "Improved non-deterministic planning by exploiting state relevance". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 22. 2012, pp. 172–180.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[12] N. Simon and C. Muise. "TattleTale: Storytelling with Planning and Large Language Models". In: *ICAPS Workshop on Scheduling and Planning Applications*. 2022.

[13] C. Purdy, X. Wang, L. He, and M. Riedl. "Predicting generated story quality with quantitative measures". In: *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2018.

[14] M. O. Riedl and V. Bulitko. "Interactive narrative: An intelligent systems approach". In: *Ai Magazine* 34.1 (2013), pp. 67–67.