Does Robin Hood Use a Lightsaber?: Automated Planning for Storytelling

Nisha Simon

School of Computing, Queen's University
Kingston, ON Canada
nisha.simon@queensu.ca

Abstract

Humans have used stories to entertain, educate, and persuade audiences for centuries. The advent of modern AI tools in the form of Large Language Models (LLMs) such as ChatGPT continues to fulfill this purpose. However, while recent work has shown that LLMs can successfully be used for narrative generation, they lack coherence and can be prone to repetition and stilted language. Automated Planning can, therefore, be combined with Natural Language text generation to create narratives (stories) that are logical, coherent, and believable. A planning model provides scaffolding to an LLM so that the LLM's language generation is context-dependent to allow users to create more coherent, logical, and believable stories in a variety of domains.

Introduction

My research's primary motivation is using Automated Planning for Storytelling. Large Language Models (LLMs) are excellent at generating output text based on input text that they have already seen, but they are still unable to reason logically about real (or imaginary) worlds in which they find themselves. A solution to this problem is to use Automated Planning, which is the logical process of thinking before acting for an *agent* or actor to progress from a given initial state to a goal state within the constraints of a specified environment or domain: here, the agent is the story-teller. This central idea is illustrated in Figure 1, where stories generated with only an LLM are shown to be incoherent and illogical, while stories that use an LLM in conjunction with a valid plan are logical and more believable.

Planning problems are represented using the Planning Domain Definition Language (PDDL) (Muise et al. 2019). Planning problems generally use two files: the *Domain* file and the *Problem* file. The Domain file consists of the *requirements, types, predicates* and *actions*, while the Problem file consists of the *objects*, the *initial state* and the *goal*. A particular domain could have multiple problems that are associated with it. The domain and problem files are fed into an automated planner as inputs in PDDL format, and the planner then produces a *plan* (typically represented by a sequence of actions or steps). The steps of the plan are fed to the LLM one at a time and the LLM then generates a story.

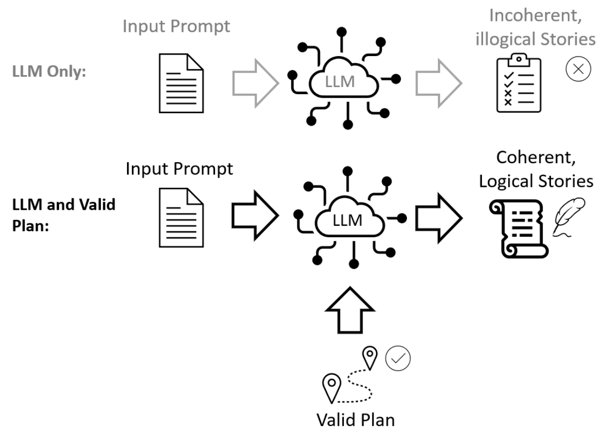

Figure 1: Effect of using a valid plan combined with an LLM

Conceptually, we represent a planning problem *P* 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 $\mathbf{a}$ in the set of Actions **A** has three characteristics: PRE(a) : the preconditions of action $\mathbf{a}$ or the set of fluents that must hold to execute action $\mathbf{a}$, DEL(a): the set of actions that are removed from the current state when action $\mathbf{a}$ is executed, or the '*delete effects*' of action $\mathbf{a}$, and ADD(a): the set of actions that are added to the current state when action $\mathbf{a}$ is executed, or the '*add effects*' of action $\mathbf{a}$. If PRE(a) $\subseteq$ s, the agent can take action $\mathbf{a}$. We *progress* from a state *s* to state s' using action $\mathbf{a}$ by removing every fluent that $\mathbf{a}$ deletes, and then adding every fluent that $\mathbf{a}$ adds. That is to say *Progress* (s,a) = (s $\setminus$ DEL(a)) $\cup$ ADD(a). The goal is achieved when G $\subseteq$ s.

Current and Past Work

My initial work included a study based on children's stories such as '*The Way Home for Wolf*' (Bright and Field 2020), '*Robin Hood and the Golden Arrow*' (San Souci and Lewis 2010), and '*The Paper Bag Princess*' (Munsch and Martchenko 1980). The input stories were selected so as to contain simple language (vocabulary) and sentence construction.

For the quantitative analysis, I determined if the generated stories met the required *character* and *author* goals. Part of Speech (PoS) tags were also used to compare the nouns and verbs present in the plan and in the generated stories. Almost all of the nouns (characters, objects, and locations) and verbs (actions) of the plan are reflected in the generated story, showing that the resulting narrative is more coherent than stories that are generated using only plain text prompts to the LLM.

I also completed an initial user evaluation study by asking participants to complete an anonymous questionnaire based on an analysis of simple children's stories. One set of stories was generated solely by an LLM, and another set of stories was generated by an LLM in conjunction with a valid plan that an Automated Planner created. Participants were unaware of which method was used to generate each set of stories. I conducted the human evaluation of the generated stories based on the method that was followed by Purdy et al. (Purdy et al. 2018). The human evaluators graded the stories generated using a Plan and an LLM higher in general on a variety of metrics than those generated only with an LLM. When asked if they agreed or disagreed with statements analyzing the quality of the generated stories, such as "*This story avoids repetition*" or "*This story's events occur in a plausible order*", evaluators judged the stories that were generated with a plan and an LLM higher than the stories that were generated only with an LLM. This preliminary work is described in the publication, "*TattleTale: Storytelling with Planning and Large Language Models*", presented at the Scheduling and Planning Applications workshop (SPARK), held in conjunction with the International Conference on Automated Planning and Scheduling (ICAPS) (Simon and Muise 2022).

Future Work

Some of the other topic extensions of my work will include further research on the most appropriate prompts for LLMs, stories about time which include time constraints in planning, as well as non-deterministic planning for '*choose-your-own-adventure*' stories.

Prompt Engineering for LLMs

LLMs maintain output coherence based on the provided plan's actions and on the initial inputs with which they are seeded. Prompt engineering is, therefore, a key element in generating believable stories. The way in which prompts are structured affects the quality of the generated output sentences. I will examine more closely the prompt content and structure that are most suitable for generating coherent, believable stories. Prompts for a story include both the actions that are provided by the output plan and also the so-called '*hidden*' or '*story-agnostic*' prompts that provide style guidelines as well as background information. The story-agnostic prompts provide background or common-sense information that guides the rest of the output but can be common to multiple stories. The story-agnostic prompts are also useful for style purposes, such as dictating how the story is written. The LLM requires at least two (or more) initial patterns of '*action*' and '*story*' prompts in order to reliably recognize and generate the text. LLMs can be considered to be '*few-shot learners*', and the '*pattern*' that they learn from is the combination of the story-agnostic prompts, as well as the initial inputs.

Stories about Time

Stories that have time constraints (Temporal Planning) are an exciting extension of classical planning. The resulting plans may describe stories where a limited amount of time is available to perform a specific action or where a certain set of actions must be performed in a particular sequence. Linear Temporal Logic (LTL) involves time considerations in the categories of '*safety*' and '*liveness*'. For instance, Batman, the alter-ego of billionaire Bruce Wayne, uses his superhero persona to fight crime in the city of Gotham. An example of a safety consideration in a story featuring the character of Batman is that "*The Batmobile must maintain an open communication channel to Alfred at the Batcave while Batman is on a mission away from Wayne Manor*". Examples of a liveness consideration are that "*The active communication channel in the Batmobile cannot be deactivated until it returns to the Batcave after a mission*", or "*Once a mission begins, Batman always captures the villain before the end of the mission*".

Choose-Your-Own-Adventure Stories

'*Choose-Your-Own-Adventure*' stories are a form of non-deterministic environments. This means that an agent's action may have an impact on the world that is not known until the time that the action is executed. For instance, Batman may decide to take the action of entering a disused, abandoned warehouse with the hope of capturing a crafty villain called the Riddler. However, until the action is taken (executed) and Batman actually enters the warehouse, it may be unclear whether or not the Riddler is lurking inside the warehouse until this decision is made by the reader.

References

Bright, R.; and Field, J. 2020. *The Way Home for Wolf*, volume 1. Scholastic Press.

Muise, C.; Chakraborti, T.; Agarwal, S.; Bajgar, O.; Chaudhary, A.; Lastras-Montano, L. A.; Ondrej, J.; Vodolan, M.; and Wiecha, C. 2019. Planning for goal-oriented dialogue systems. *arXiv preprint arXiv:1910.08137*.

Munsch, R.; and Martchenko, M. 1980. *The Paper Bag Princess*, volume 1. Annick Press.

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

San Souci, R. D.; and Lewis, E. 2010. *Robin Hood and the Golden Arrow*, volume 1. Scholastic Press.

Simon, N.; and Muise, C. 2022. TattleTale: Storytelling with Planning and Large Language Models. In *ICAPS Workshop on Scheduling and Planning Applications*.