HaLLMark: Innovative Story Driven LLM Game (supporting document)

Jiří Macháček, Radek Richtr, Jan Matoušek, Ondřej Brém

Abstract

The game HaLLMark explores the innovative use of large language models in interactive storytelling. Developed using the Unity game engine, this game allows players to engage in dynamic and coherent conversations with in-game characters, who in turn generate and narrate compelling stories. Our research involved analyzing the currently top-rated large language models and evaluating their practical applications in story generation and character impersonation. Based on this analysis, we implemented a playable prototype designed to provide valuable feedback on the quality of stories generated by various models while also delivering a unique gaming experience. The results demonstrate that language models can effectively create consistent narratives, paving the way for new possibilities in game development. Testers have praised the game for its innovative concept, its ability to maintain a cohesive storyline, and its adherence to traditional three-act story structures.

HaLLMark represents a significant advancement in the use of artificial intelligence within the realm of narrative-driven games. Traditional story-driven games often limit player interaction with non-playable characters (NPCs) to predefined sets of actions and dialogue choices. HaLLMark proposes a breakthrough approach, utilizing state-of-the-art large language models to facilitate freeform interaction with NPCs, enabling them to generate and narrate immersive, coherent stories in real-time.

The practical outcome of the development is the HaLLMark prototype, implemented in Unity, which serves as a testbed for evaluating various language models. The game not only allows for comparative analysis based on the quality of generated stories and character behaviors but also provides a novel gameplay experience. Through user testing, we assessed the prototype's effectiveness in utilizing different language models, establishing criteria for their evaluation, and identifying potential improvements and extensions. The game's narrative structure adheres to the traditional three-act framework, ensuring a familiar and engaging story progression. Moreover, the generated stories are consistently coherent, resistant to hallucinations, and robust against intentional prompt injection attacks, providing a seamless and immersive user experience. The playable prototype named HaLLMark is a 2D strategy game inspired by Reigns and

Crusader Kings. Powered by large language models, the game performs various NLP tasks from story generation to quest completion decisions. It targets players who enjoy creating their own stories and strategy enthusiasts. In HaLLMark, players act as the king of a fictional fantasy kingdom, resolving issues for their subjects. Players communicate freely with in-game characters via text, aiming to maintain four metrics above certain levels: kingdom's wealth, strength, king's popularity, and court's patience.

1 Game

The player can freely communicate with game characters using simple text, and the goal is to maintain four metrics above a certain level for a sufficient amount of time – the kingdom's wealth, the kingdom's strength, the king's popularity, and the court's patience. The court's patience is a special metric that significantly decreases in response to the player's deviation from the royal role or nonsensical decisions. The metrics will range from 0 to 100 and will be continuously adjusted.

If any metric is completely exhausted, the player is dethroned and loses. Surviving for 5 days (game rounds) results in a win and a future outline of the kingdom. The game follows a three-act structure, increasing difficulty and tension as the story progresses.

Players choose their name, kingdom's name, and main advisor (court jester, marshal, or chief spy) at the start. Based on these choices, kingdom information is generated. Players can interact with three main characters in the throne room: the advisor, a subject seeking an audience, and the chancellor. Three additional characters – a peasant, priestess, and nobleman – comment on decisions.

The game includes a royal journal summarizing daily events and a notebook with important story information. Players can optionally complete tasks, typically involving specific decisions or using certain words in dialogues. These tasks enhance the player's decision-making options as a king.

Each round represents a day at court, starting with a subject presenting an issue. Players can ask up to four questions before deciding. The consequences of decisions adjust kingdom metrics and update the royal journal and story notebook. Important story information is stored in a vector database to select relevant data for NLP tasks.

The game takes place entirely in the throne room, with stylized graphics for characters, backgrounds, and UI elements. It requires a constant internet connection but is not hardware-intensive.

Figure 1: Tutorial

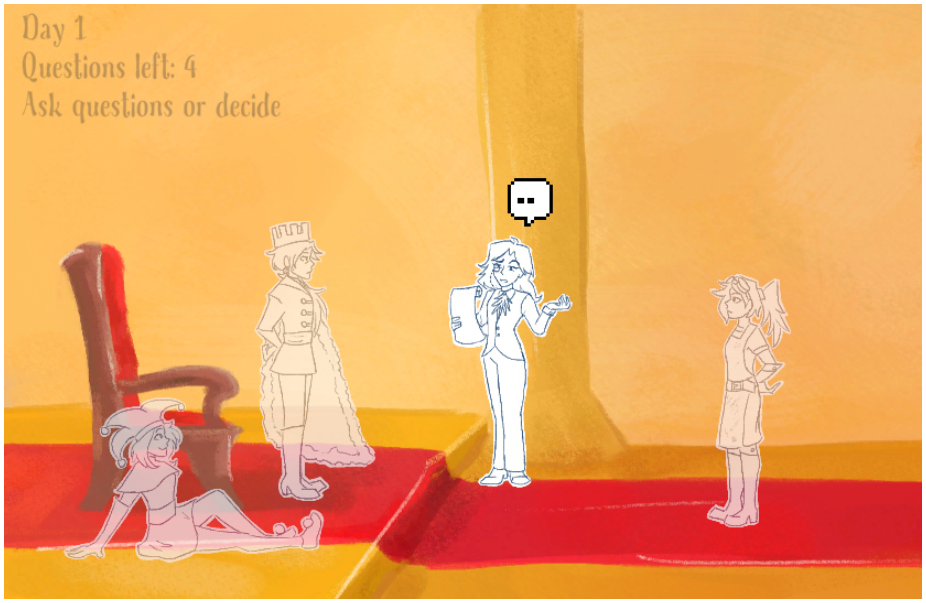

Figure 2: Waiting for an answer

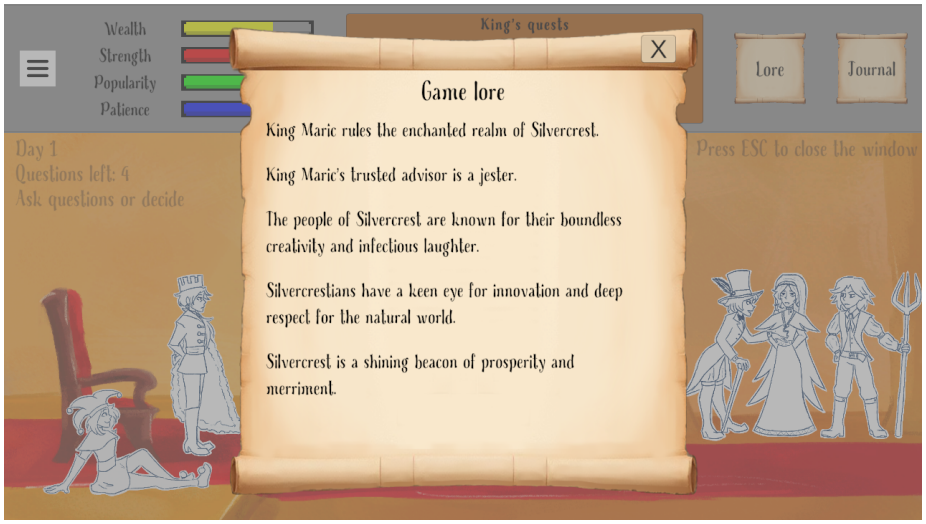

Figure 3: Generated lore

Figure 4: Full window example

2 Existing Large Language Models

The presented game was tested with many LLM models but only the best language models according to the ranking on the website [3] will be mentioned. The quality of the language models mentioned here has thus been verified by users worldwide.

In the mentioned ranking, language models are sorted based on how they perform in chatting with users. Since chatting is the main functionality of freely available language models, this evaluation metric is considered valid, among other things, in the practical part of this work. Language models do not typically support specific functionality for story generation, and through chatting with a language model, we can get closest to the expected functionality. When it comes to simulating a dialogue with a fictional character, chatting with a language model is the best option to utilize the model for this task.

If one company has multiple models in the mentioned ranking, only the one with the highest Arena Elo metric is further discussed. The models are subsequently sorted according to this value. All the following information from the websites [3] is sourced as of April 12, 2024. As of this date, over 650,000 votes had been submitted to the language models on the site.

2.1 GPT-4 Turbo

The language model gpt-4-turbo-2024-04-09, which ranked best among all the language models developed by OpenAI, was released on April 9, 2024, as the latest model powering the GPT-4 Turbo endpoint [4].

Its knowledge of real events extends to December 2023. Details about the training of the GPT-4 Turbo language model are unknown, but the training procedure is likely not different from the previously introduced GPT-4 language model. The training data of the GPT-4 model consist of freely available data on the internet and third-party data. Content that violates OpenAI's policies was filtered out of the training data. The GPT-4 model subsequently underwent fine-tuning using RLHF [5].

2.2 Claude 3 Opus

The Claude 3 language models – Haiku, Sonnet, and Opus – were introduced on March 4, 2024. These are currently the most advanced language models created by Anthropic.

The models are trained on data freely available on the internet, reaching up to August 2023. The training data also include data generated by Anthropic and third-party data. To improve the quality of the training data, the data were filtered and cleaned several times. The Claude 3 models were pre-trained on the data and then exposed to the RLHF process. The intended use of the Claude 3 family language models is to solve NLP tasks. In addition to text inputs, the models can accept images and documents. The language models in this family do not have internet access. The strength and cost of individual models increase

in the order mentioned here. Claude 3 Opus is therefore the most expensive but also the most powerful model, and hence only it will be considered from the entire family [6].

2.3 Gemini Pro

Since December 13, 2023, language models from the Gemini 1 family – Gemini Nano, Gemini Pro, and Gemini Ultra – have been available to developers. The Gemini Pro language model ranked highest in the freely available chat platform by Google, replacing the Bard language model [3, 7].

Google is the only company providing details about the architecture of their language models. The Gemini models consist of transformer decoders. A specific decoder is chosen based on the type of output the language model is to generate [1].

The Gemini Pro language model currently has internet access, allowing it to work with the most up-to-date information. The Gemini language model was pre-trained on data from web documents, books, and code. Filters were applied to the training dataset to enhance its quality and remove potentially harmful content. The language model was fine-tuned using instruction tuning and RLHF [1].

Language models developed by Google cannot currently be used via official API from the European Union states.

2.4 Command R+

The Command R+ language model was released on April 4, 2024, by the Canadian company Cohere. It is a language model optimized for use with the RAG paradigm, trained to handle loads typical for large enterprises. The language model is open-source and can be fine-tuned. It is the only mentioned language model that has published its number of parameters – 104 billion [8, 2].

The knowledge of the Command R+ language model ends in March 2024. Command R+ excels in communication in the 10 most widely used languages globally, but its training data also included Czech texts [3, 2].

2.5 Mistral Large

The Mistral Large language model was released on February 26, 2024, and at the time of its release, according to Mistral AI [9], it was the second-best freely available language model via API (GPT-4 is considered the best). At the time of its release, it introduced a relatively unprecedented high token limit in the request, namely 32,000 tokens. The Mistral Large language model can fluently speak in 5 languages – English, French, Italian, German, and Spanish.

Table 1: The table consists of the range of knowledge, meaning the date of the latest data used to train the language models that was used fot HaLLMark testing. Model Gemini Pro is connected to the internet and has access to current information. Listed are also the results in the Hellaswag and WinoGrande benchmarks. The Gemini Pro model has API availability limited to the European Union. Data for individual columns were obtained from source.

Language Model	Open-source	Available API	Latest Data	Hellaswag [%]	WinoGrande [%]
GPT-4 Turbo	−	+	March 2023	95.3	87.5
Claude 3 Opus	−	+	April 2023	95.4	88.5
Gemini Pro	−	−	May 2023	84.7	unknown
Command R+	+	+	October 2023	88.6	85.4
Mistral Large	−	+	unknown	89.2	86.7

Table 2: The table shows whether each language model contains functions that are widely available through APIs, i.e., automated access to the model, document support, handling a large number of embedded tokens, and JSON mode. Each of these functionalities can be turned off, meaning it is either enabled or not.

Language Model	Volatile Memory	Streamed Responses	Text Embeddings	Search	JSON Mode
GPT-4 Turbo	+	+	+	+	+
Claude 3 Opus	+	+	−	−	−
Gemini Pro	+	+	+	−	−
Command R+	+	+	+	+	−
Mistral Large	+	+	+	+	−

Table 3: For each model, the cost per 1 million tokens passed through the API is given. Additionally, the input and output limits are provided. Note that for models with token limits, they represent the sum of input and output tokens. The Gemini and Command R+ models, although based on the transformer architecture, do not have token limits. This is because these models are able to generate relatively coherent text for longer tokens.

Language Model	Cost per Input ($/1 mill. tokens)	Maximum Input (tokens)	Cost per Output ($/1 mill. tokens)	Maximum Output (tokens)
GPT-4 Turbo	10	128,000	30	4,096
Claude 3 Opus	15	200,000	75	4,096
Gemini Pro	0.625	30,720	1.875	2,048
Command R+	3	128,000	15	Not provided
Mistral Large	8	32,000	24	Not provided

3 Game Using

download link link

API keys For best performance, we recommend you to use your own API keys (LLM model + Pinecone database).

Keys are located in `\HaLLMark\_Data\StreamingAssets` in the file api-keys.json. For playing at least one LLM API and Pinecone database must be admissible.

References

[1] GEMINI TEAM. Gemini: A Family of Highly Capable Multimodal Models [online]. 2023. [cit. 2024-03-19]. Available from arXiv: 2312.11805 [cs.CL].

[2] USTUN, Ahmet; HOOKER, Sara; ASHIMINE, Ikko Eltociear. Model Card for C4AI Command R+ [online]. 2024-04. [cit. 2024-04-12]. Available from: `https://huggingface.co/CohereForAI/c4ai-command-r-plus`.

[3] LARGE MODEL SYSTEMS ORGANIZATION. LMSYS Chatbot Arena Leaderboard [online]. 2024. [cit. 2024-04-07]. Available from: `https://chat.lmsys.org/`.

[4] OPENAI. New models and developer products announced at DevDay [online]. 2023-11. [cit. 2024-04-07]. Available from: `https://openai.com/blog/new-models-and-developer-products-announced-at-devday`.

[5] OPENAI (2023). GPT-4 Technical Report [online]. 2024. [cit. 2024-04-08]. Available from arXiv: 2303.08774 [cs.CL].

[6] ANTHROPIC. The Claude 3 Model Family: Opus, Sonnet, Haiku [online]. 2024. [cit. 2024-04-07]. Available from: `https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf`.

[7] PICHAI, Sundar; HASSABIS, Demis. Introducing Gemini: our largest and most capable AI model [online]. 2023-12. [cit. 2024-04-07]. Available from: `https://blog.google/technology/ai/google-gemini-ai`.

[8] GOMEZ, Aidan. Introducing Command R+: A Scalable LLM Built for Business [online]. 2024-04. [cit. 2024-04-12]. Available from: `https://txt.cohere.com/command-r-plus-microsoft-azure/`.

[9] MISTRAL AI TEAM. Au Large [online]. 2024-02. [cit. 2024-04-08]. Available from: `https://mistral.ai/news/mistral-large/`.