CSCI 5541 Progress Report (*Tired Tokenizers*)
Lairs and Language Models: Tabletop Gaming Narrative Assistant

Nicholas Padilla, Jack LeGeault, and **Jacob Cadavez**
padil014@umn.edu
legea008@umn.edu
cadav006@umn.edu
Project Website: https://padpy.github.io/csci5541-project-template/
Github: https://github.com/padpy/VoloLLM

Abstract

VoloLLM is a Dungeon and Dragons dungeon master chatbot. It is designed to ingest Dungeons and Dragons adventure modules as guide a single player through the adventure.

We have also created a Project Website for readers to view as a supplementary resource.

1 Introduction

Dungeons & Dragons (D&D) is an open-ended tabletop role-playing game. Players take the role of a single crafted character, while the Dungeon Master (DM) creates a story, following conventions from a campaign book, and will lead the players through it. DM work can get extremely tedious and taxing as they have to deal with complex intertwined storylines, setting and character consistency, and crafting engaging gameplay. While enjoyable, many players can find it challenging to schedule player sessions over multiple weeks, coordinates groups of 4 or more people, and for DMs to prepare thoroughly for each session.

Our team aims to develop an application that will take the role of a DM and will guide players through an predefined story, adventure modules, and be able to respond to players' input and actions while building a cohesive story. Our team hopes to address the issue of potential players not being able to find others who have the time or energy to join them on a session.

Role playing is also a popular genre of computer games. The ability to interact with the world and story in creative and expressive ways is the root of enjoyment millions of video game players. Our research has interesting potential applications for video games, opening players up to more interactive and dynamic story telling.

2 Background

The use of LLMs to drive narratives, generated stories, and agents is an active field of research. There are many existing techniques for LLM narrative generation and storytelling, but all in insolation. Our team analyzed and researched the effectiveness of said techniques and how they could be potentially implemented in our project.

2.1 Fine-tuning LLMs

Fine-tuning has been shown to improve storytelling (Sun et al., 2023) abilities of an LLM, enhancing the stylistic capabilities and showing writing prowess (See et al., 2019) when given specific prompts and requirements (Gite et al., 2024). Additionally, when utilizing parameter-efficient techniques such as QLoRA, the model does not require a high volume of stories to feed into its training process (Dettmers et al., 2023).

Although the stylization of the generated stories increase (Jeong, 2024), an ethical issue arises of how to deal with plagiarism and stealing of authors' works (Kapania et al., 2024). There have been studies highlighting cases where models have written eerily similar stories to preexisting ones (Rosen, 2023), and unless safeguards are implemented, there is a high risk of real life stories being stolen and published as fully original (Xie et al., 2023).

Another downside to fine-tuning is that this technique has not been utilized to improve storytelling consistency and memory (Guan et al., 2021).

2.2 Retrieval-Augmented Generation (RAG)

RAG techniques retrieve relevant story elements and maintain contextual relevance depending on the input by the player (Gupta et al., 2024). Other researchers have found that by utilizing RAG can enhance plot line consistency and have the ability to incorporate 'common-sense' constraints, meaning the LLM has a rationale for every line given and doesn't hallucinate (Gao et al., 2024; Wen et al., 2023). In the studies found, RAG based LLMs generated stories that have been evaluated by human annotators to be the most creative and complex in comparison to other prompting techniques (Wen et al., 2023). It also mitigates the problem of plagiarism as it has been found that the generated stories have a zero N-gram overlap with original text (Wen et al., 2023).

While effective for static or partially structured narratives, this method is not designed for the dynamic, player-driven nature of D&D campaigns (Gupta et al., 2024).

2.3 Prompting Techniques

In story generation tasks that require a higher level of consistency and baseline rules, a story planning methodology is preferred (Xie et al., 2024), where events are built before the story is generated. Pretrained language models (PLM) (See et al., 2019) typically generate a storyline that will guide the generation process (Liu et al., 2021; Brown et al., 2020). From previous research, many have found that the coherence and fluency ratings were increased by utilizing another model to aid as this pre-processed event generator (Xie et al., 2024).

Often, story generation system architectures are setup in a manner where the original text is first broken up up into smaller problems and plot lines into a 'planner' (Wang and Kreminski, 2024). When needed to describe a certain scenario, the information for a specific plot line from the 'planner' is injected inside a prompt, helping provide baseline rules and context (Simon and Muise, 2024).

3 Motivation

Our project aims to provide a guided narrative experience for tabletop role playing (TTRP) gamers. There are many new and classic adventure modules that gamers would love to play but may not have the time or people to play with. To help lower the barrier of entry, we made a AI system that is able to accomplish three main tasks that are typically required by a dungeon master: respond to users in-game actions in a creative and engaging manner, account for users' past actions, and move users through a structured narrative.

To accomplish this, we will leverage the human-like text generation of LLM assisted by modern prompting techniques where required, leveraging Retrieval Augmented Generation to improve retrieval of relevant content from the module and past actions, and use a novel story progression system that automatically generates narrative milestones from source texts and tracks story progress.

This combination of features elevates the improvisational nature of LLM in the DM role. Utilizing both a RAG and story progression system mean that users can still experience enjoyable and creative play sessions that stay faithful to the original source material.

4 Approach

For our project we created an end-to-end DM system called VoloLLM. This system is able to ingest a Dungeons and Dragons PDF adventure module and create a database and populate the contents of a progression system. Each session begins with a predefined message to kick start the adventure. From there players and the chatbot collaboratively build a narrative. Below details the major components of this approach.

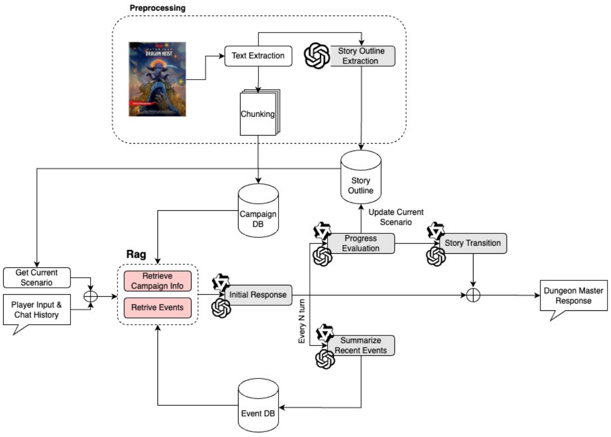


Figure 1: Diagram showing the pipelines and flow of what data and prompts are passed to make VoloLLM generate a response.

Technology	Usage
Qwen 2.5 / ChatGPT 4o	Run-time LLM
ChatGPT-o1	Preprocessing LLM
Typescript Web App	Frontend

Table 1: This table contains the technologies and frameworks used to develop VoloLLM.

4.1 Campaign Preprocessing

While we could create a system that is creates narratives entirely improvisationally, there is a wealth of existing official and unofficial adventures that players may want to play as they have existing rules, settings, and plot lines. Because of this we need of a way for out system to ingest these stories and automatically process Dungeons and Dragon adventure module. To initiate preprocessing, we need a source adventure book in PDF format. The PDF is then converted to plain text.

A database is then populated to contain all the data from the adventure module, called the Campaign Database. This database contain documents that are 1000 character chunks of text from the adventure manual with 200 characters of overlap between documents. The Campaign Database is a HuggingFace in-memory vector store (LangChain, a), that stores the documents and associates them with a vector embedding for retrieval.

The story progression system also needs the adventure module to be summarized and stored for future story reference. This is accomplished by generating a sequence of scenarios off the module, and storing it all in a JSON file. We will describe this process in more detail in *Story Progress Evaluation*, but note that for the purposes of this project, we manually restricted the story summarization to the first chapter of the module (Deepwater). Our system can automatically perform this task with the entirety of any campaign module.

4.2 Action Summarization & Database

In a Dungeons and Dragons campaign, one of the most important gameplay components is the feeling that your unique choices make an impact on the story. To do this, text summarization is performed on a window of chat history, meaning the user's actions will be saved and remembered. The LLM is then asked to extract events in that occur during this window, and store them in a HuggingFace in-memory vector store for later retrieval. This vector store is known as the Action Database.

For our implementation, a summary is generated every 3 turns, where each turn consists of the 1 human message and 1 AI response. The chat history window is 5 turns in length. This window was determined through trial-and-error and human qualitative evaluation. The chat history window of 5 provided the an adequate window of context for the smaller models to maintain a coherence in the flow of conversation and events without inducing negative side-effects. We found that if the chat history was much longer the model tended to repeat sentences or phrases for the chat history. We choose to update the player action history every 3 turns, so there is overlapping context between summations.

4.3 Retrieval Augmented Generation

To create a cohesive response, a DM must be aware of facts about the world, and be aware of actions the player has taken. Rather than fine-tune the model with this information, Retrieval Augmented Generation (RAG) provides a method of retrieving information from a data store and then injecting into the prompt to add context and lead to better quality and more accurate model responses (Gao et al., 2024). We have two sources that we use for retrieval augmented generation: the campaign database and the action database. The former contains the information of about the world and story, and the latter contains information about choices that the player has made over the course of their campaign.

For both retrieval systems, we use the Hugging Face *InMemoryVectorStore* (LangChain, a). This provides us with an simple way to create databases from document collections and to inject new documents into the database as they are generated for the action database. BERT, bert-base-uncased, (HuggingFace) is used to generate vector embeddings for document comparison and retrieval.

4.4 Story Progress Evaluation

The progress evaluation system used in this project is the piece of novel contribution from our team. Although story information can be retrieved from the adventure module, there is no enforcement or guidance mechanism that moves players through the story. RAG context retrieval improved consistency in the game environment and characters, but through play testing, we determined this was insufficient for consistently progressing the story. To more consistently move players through the campaign we implemented a progress evaluation system that is aware of the intended path of the player through the story and the means by which the LLM can progress the player along this path.

During preprocessing, we initially convert the PDF adventure module into a plain text file. We then extract the first chapter of the campaign manually. We then pass the first chapter to ChatGPT o1 (OpenAI, b), and ask it to do two things: summarize the each story point in the module into scenario descriptions and then determine multiple example resolutions criteria for these scenarios. Adventure modules are particularly well suited for such a task, because the are intentionally structured in such a way that events in the store are sequentially ordered in each chapter and information about the event outcomes is listed in each section. These events and resolution criteria are numerically ordered and placed on a linear progression path.

At the beginning of each new session, the players current scenario is set to 1, the first event in the module. The player then plays the scenario out as they chose, with the current scenario included in the prompt. With each player message, the LLM generates the initial response. Based on the response and the chat history the LLM is asked to decide if the scenario is resolved and the story is ready to progress. If the scenario is not yet resolved, then nothing happens, but if it is resolved, we prompt the LLM to generate a new response based on the initial response and the description of the next scenario.

Additionally, the current scenario number is incremented by one. Until the next step in story progression, the current scenario is included as context for the LLM during response generation.

4.5 Models Used

For the evaluation of the the approach, we used three LLM models OpenAI's ChatGPT 4o (OpenAI, a), Alibaba's Qwen 2.5 14B, and Qwen 2.5 7B (HuggingFace, 2023). ChatGPT 4o is a paid API that allows access via and API key, and the Qwen models are open source, and ran locally on a server. Although, it is unknown how many parameters ChatGPT 4o has, we used it as a performance benchmark to compare locally deployed models as we are operated under the assumption that is generally considered more powerful than Qwen 2.5 14B & 7B and is tuned for general purpose prompting tasks.

The purpose of running the local models is that these make the ability to deploy VoloLLM to the public easier and more cheaply. By using local models, there is no need to pay API cost, worry about API key distribution, and users can further refine the models to improve experience. We wish to gauge the viability of using local model considering the potential benefits to consumers.

4.5.1 Website and User Input

Our DM chatbot needs an interface for users to interact with. To accomplish this, we created a simple TypeScript interface that handles all signaling to the LLM inference API and text response display. The interface also supports the the new game creation, user data entry, and character information entry. A screenshot of the app can be seen in Figure 2 in the appendix.

5 Development Challenges

There were a number of challenges that we encountered during the project. Firstly, we intended to fine-tune the model to improve the style and format of the responses. We initially planned to do this using transcripts from a a Dungeons and Dragons podcast, but we found the model seem to overfit to the dataset and would often respond with mannerisms of the podcast's DM, and included podcast characters in responses. We explored using synthetic data generated using ChatGPT, however, because of the large response sizes needed to emulated the chat history and other contextual information, the cost was prohibitively expensive.

We found that improved prompt construction using techniques learned through the course of the semester greatly improve response quality and consistency. Particularly, for the local models techniques such as automatic chain of thought prompting and few-shot prompting improve greatly improved the detail of response and output format.

Our initial project idea also assumed that RAG would be sufficient for progressing the story for users, but this proved to be inconsistent at progressing the story. This is why we investigated that mechanisms for story progression. Other methods we have detailed above typically set story points much as our system does, and set constraints for action or story telling. However, none these methods above provide a means for metering use progress and pushing them along the narrative.

The most persistent issue experience throughout the project was generation of consistent output formats. Because so many systems rely on LLMs to generate story progress status of event summaries, the system is vulnerable to issues of prompt structure leaking into the generate response when it is parse from the full LLM response. For instance if the "CHAT HISTORY" heading string leaks through the LLM response, then it is can end up in chat history or action history. This can then lead to the string appearing event more frequently and leading to all further responses to be ill-formatted and difficult for a human to read.

Response formatting issues are most prevalent on local models, but also occur with ChatGPT models with less frequency. The addition of output formatting and parsing with LangChain (LangChain, b), the issue still persisted.

6 Experiments

The are three aspects that we wish to evaluate in our report: user satisfactions, model size and its impact on play quality, and story alignment with source material. Below we detail the quantitative and qualitative methods we pursue to evaluate each.

6.1 Player Satisfaction

We wanted to measure the overall quality of player experience, and to do this we are using a modified version of the GUESS 18 survey. GUESS 18 (Phillips et al., 2016) is a video game satisfaction psychometric evaluation survey that tracks 9 aspects of player satisfaction: usability/playability, narratives, play engrossment, enjoyment, creative freedom, audio aesthetics, visual aesthetics, personal gratification, and social connection. Players rate their agreement with a prompt on a 1 to 7 scale with 1 meaning *Strongly Disagree* and 7 meaning *Strongly Agree*. Given that our application is a single player text-based game usability, audio, visual, and social connection do not applying. However, we modify this these questions by combine the audio and visual aesthetic questions into a textual aesthetic category, social connection questions were modified by asking if users would enjoy the game if they could play the game with others, and the usability questions were excluded.

All surveys were conducted through the use of Google Forms, and conducted after user played the game for at least 15 minutes. A link to the form was provided to users through the games chat interface 3.

6.2 Model Selection and Play Quality Impact

To provide the best user experience, it is important to select a model that provides the highest quality responses. To accomplish this, our team investigated three different models to evaluate their quality: OpenAI ChatGPT 4o, Qwen 2.5 14B, and Qwen 2.5 7B. We determine model quality by having members of the research team using each model in play sessions, then recording player satisfaction scores based on the modified GUESS-18 scoring system.

Due to limited resources, we chose to conduct this testing entirely within the team. This is due to human capital, budget, and technical constraints. We acknowledge the statistical limitation of such an approach, but we feel that it is sufficient for evaluating model performance.

6.3 Story Alignment

The project aims to allow users to play through existing stories from an extensive catalog of existing adventure modules. If players are going to play through these predefined stories, there must be some measure of alignment between the generated story and the source material. To measure this alignment, we will take the entire text of the first chapter of the adventure module and use Corpus BLEU and Rouge scores to measure how closely the generated text aligns with it.

To validate the story line created for the story progression system, we chose to manually extract the section of the adventure module that should line up with the scenario descriptions and use cosine similarity to compare an embeddings generated from the manually extracted section and the automatically generated scenario description.

7 Results

7.1 Player Satisfaction

In total 15 players, including the 3 researchers, played VoloLLM chatbot with the OpenAI LLM backend. From player responses, all categories of player satisfaction rated neutral or above. From Table 5 the two highest rated categories of satisfaction were *Creative Freedom* and *Enjoyment*. Players reported surprise at how capable the chatbot was at addressing their character choices. They felt that they are free to choose their own path through the story and the chatbot would smoothly accommodated almost any choice while incorporating choice details.

The two lowest categories of satisfaction are *Player Engrossment* and *Personal Gratification*. Some of the most cited complaints by players was the lack of challenge in the story. Player sited that the bot was too compliant to the point of taking away for story enjoyment. During one anecdote a user was able to use sci-fi weapons to defeat their enemies, or in an other instance the user stopped performing the quest and instead went to go get burgers. The lack of challenge and constraint makes player feel unchallenged and unsatisfied.

Additionally, the chatbot responds with relatively large blocks of text reaching 256 characters in length. Users that do not enjoying reading reported that the text-based nature of the game made the experience less enjoyable. This may provide an explanation for the results in Table 4 where the greatest degree of standard deviation in both questions is *Player Engrossment*, as *Question 6* which asks about player boredom.

7.2 Model Selection and Play Quality

When comparing the quality of play between the locally run models, Qwen 2.5 7B & 14B, and the OpenAI the clear winner is OpenAI. It is the only model that performed above neutral for all categories of satisfaction (Table 6) and satisfaction questions, except for *Question 6* where it was reported as being more boring than the local models. The one area where the Qwen models are reported beating ChatGPT 4o in our evaluation is *Creative Freedom*. From our experience, Qwen models tend to more directly address actions the player takes, and with shorter responses. These short, direct messages can make the generated responses feel are more attentive to user choice, because user actions are reflected in more of the generated text in absolute terms.

The category of satisfaction where the local models most clearly fail is the *Textual Aesthetics*. These models have the highest tendency to leak portions of the system message into the response. For example, there are situations where the chat history repeated as part of the generated response. Even after output formatting and parsing with LangChain, this leakage pollutes response. This leads to an overall negative aesthetic experience for users and can cause the LLM to go into a loop, where regardless of what is typed by the user, the LLM repeats the chat history as the response.

Category	GPT 4o	Qwen 14B	Qwen 7B
Narratives	**6**	5.33	5.33
Player Engrossment	**4.67**	4	2.67
Enjoyment	**4.5**	**4.5**	**4.5**
Creative Freedom	5.5	**5.83**	5.17
Personal Gratification	**5**	4	4.33
Social Connectivity	**6**	4	4.17
Textual Aesthetics	**6.17**	3.33	2.83

Table 2: Results from a modified Guess-18 Survey utilizing different models.

7.3 Story Progress & Alignment

7.4 Story Progression

According to Table 9, only 2 players did not at least reach scenario 2 during their play through, even though they were the group of players on average that had the most number of turns taken of 32 turns, with the next highest being players who reached scenario 4 with 22.5 turns on average. This is indicative of players that "escaped" the story system. For instance, one of the users chose to immediately leave the starting scenario and get a burger, interrupting the flow of the story.

Looking at Table 8 in the appendix we can start by looking at the BLEU scores averaged across all generated responses compared to the text content of the first chapter of the adventure module. The BLEU score represents precision, or the level at with the n-grams in the generated messages occur in the chapter 1 text. We see that the BLEU1 and BLEU2 score are relatively uniform across the chats with different levels of progression through the story. This indicates that the LLM is retrieving relatively uniforms amounts of information when generating responses, regardless of progression.

Our team also evaluated Rogue scores, which represent the recall of the generated messages, to observe the total information the chapter 1 text that is being included in the generated messages. Looking at the Rouge score over the entirety of the chat, our team noticed that the total information captured from the chapter 1 text in the generated messages is relatively uniform.

However, when calculating the Rouge scores average when divided by the number of turns of play, this distribution is not uniform. Our team believed it is important to observe the rate at which the total information from the chapter 1 text is accumulating in the chat history. We see that the players that do not progress past the first scenario only have half the rate of information acquisition as players that have some story progress. This indicates that progressing player are receiving more of the story information per generated response. When we visually inspect the chat history, we also confirm that when player are progressed the generated text match the story scenarios from the story outline.

Looking at Table 8, we observed that the two users that finished playing at scenario 3, had the highest BLEU and ROUGE score averages. On closer inspection, these users only played games 15 turns in length. They appear to have progressed through the story incredibly efficiently, and this high score is not related to which scenario to stopped playing in. We believe this outlier result is simply a result of the small sample size.

7.4.1 Cosine Similarity Score

Cosine similar comparisons are performed to find how similar the embeddings of one text is to another (Lahitani et al., 2016). This method includes converting a given text into vectors and only compares the magnitude (Steck et al., 2024), disregarding the magnitude - improving decrease any bias related to text length. As a result, the returned score will improve our understanding of the semantic and meaning similarity between two messages (Januzaj and Luma, 2022).

For our model's purposes, we have decided to use cosine similarity to help gauge how similar the generated events are compared to the original campaign module, higher score indicating a closer match (Table 7). For more accurate results, the passage in the original campaign module that corresponds will be compared to with the generated response.

From our results, our team found that all of the event descriptions ranged from a moderate to high score (0.57 to 0.86), indicating high similarity to the original campaign module. This is what our team was aiming for as the closer the generated timeline events are to the original text, the higher likelihood that the generated responses using these event contexts will be accurate as well.

8 Discussion

8.1 Replicability

Our research result are focused largely on the psychometric analysis of 15 players that are playing VoloLLM for brief periods of time, 15 to 60 minutes. Players were selected from among family and friends of the authors of this report. Given the small sample size and flawed methodology for gathering impartial participants it is likely that future research based on this work may find significantly different results. This is all to say that we do not claim that our results are statistically significant or robust. However, we defend our method of sampling citing the scope of the course project and financial cost contacting significant number of participants from relevant communities, such as the tabletop gaming community, and API costs associated with running these trials.

If someone did want to run trials to replicate our findings, all source code is currently phallically available on Github: https://github.com/padpy/VoloLLM. The only additional materials need to run the code is an OpenAI API key, for access to the paid API. We have also included a copy of all questions of the modified GUESS 18 questionnaire (3).

8.2 Ethics

The main concern around our application is the generation of sensitive or immoral content. We do not implement any user safety mechanisms that protect users from inappropriate generated content, and solely rely on safety trained in the model or mechanisms implented through OpenAIs API. Our application is sensitive to injection techniques and jailbreaking. One common method for jailbreaking LLMs is to ask them to speak from another entity, bypassing their built-in safety rules. Our application is already asking the AI model to roleplay be default, making it more prone to produce otherwise prohibited contents, even through OpenAI's API.

In play testing there were several incidents that ovvured that typically would not be allowed through through the OpenAI API. Firstly, we were able to generate suggestive content during romance attempts between player characters and non-player characters. Secondly, we were about to prompt the LLM is such a way that is described graphic acts of violence towards humans and animals. This typically is not allowed by OpenAI, but in the course a play testing it freely generate this content. We can already see that, the model is prone to bypassing safety restrictions placed on the model by OpenAI.

Currently, we as the developers have access to the chat logs and data of all chats sent through VoloLLM for purposeful analysis. This may violate the privacy concerns of some users, especially when expecting a private session, as they want their chats to remain anonymous and inaccessible to other individuals. While we explicitly made this fact clear in this iteration, through a user-informed consent protocol, future interactions should also keep this ethical concern in mind and take further precautions in accessing and using this data.

8.3 Limitations

8.3.1 Handling of Complex Interactions

The story progression system provides a mechanism for the story to progress. However, progressing from one scenario to another in the story requires you meet conditions for progressing on the previous. However, this opens up possible issues with players "exploring" outside the bounds of the story and never meeting the criteria for progression. For example, the first scenario in the adventure is a fight in a tavern that the player is expected to help in or let settle by itself. One player chose to hide, but before the LLM could progress the story to allow the scenario to resolve itself, the player got became interested in the food offer at the tavern and eventually left the tavern to get a hamburger. This lead the player to go off and work for the mayor. They played over 40 turns without resolving the first scenario, because the narrative diverted to greatly for the scenario to be marked as resolved.

An alternative to this linear progression system that we have implemented the game world in a knowledge graph. All entities and their associated properties and relationships are stored as node, node properties, or edges on the graph. The events of the story are placed in this graph, and with each action the state of the story can be updated based on player action and LLM generated responses. There can also be a global update mechanism that can grab nodes that are relevant to the current scenario and update then regardless of player action. So if the user is not actively engaged with a component of the story it can still resolve without player input.

8.3.2 Adherence to Other D&D Features

The AI lacks mechanisms to ensure that player inputs and actions align with D&D core mechanics and rules, resulting in players executing actions that would not be feasible in a standard D&D setting. Furthermore, the system does not account for player classes or stats, nor does it include combat functionality. While these omissions were not a primary limitation, given this project's goal was to mimick a Dungeon Master, users may expect these features as they are fundamental to any D&D campaign. Consequently, the absence of these elements creates a misalignment with typical D&D experiences and may hinder user engagement.

8.4 Lack of Challenge

One of the most important tasks of a DM is to maintain the challenge and narrative tension to keep players engaged. The VoloLLM currently has no mechanisms for introducing this narrative tension. The system is overly-compliant user requests, which can lead to very creative outcomes but several users reported feeling unchallenged and less satisfied. Either through improved prompting or fine-tuning, VoloLLM needs to be able to restrain user choice through skill checks and unintended story consequences.

8.5 Future Research

8.5.1 Stylization

The stylization of the LLM's responses could be improved to enhance user engagement, as the current format may come across as bland, detracting from the overall experience. Additionally, the organization and readability of the chatbot's output should be evaluated by exploring and testing alternative formatting options. For instance, distinguishing dialogue from descriptive text, and differentiating in-game text from out-of-game interactions (such as clarifying questions), could provide a clearer structure. These changes may improve readability, reduce monotony, and make the content feel less overwhelming. It is recommended to test these modifications against the current iteration to assess their impact on user survey responses.

8.5.2 Surveying More Players

Using the modified Guess-18 questionnaire, future research should aim to interview a significantly larger pool of players to better identify weaknesses within the gameplay loop, as this study's sample size was relatively small. Additionally, future questionnaires should include questions about participants' prior experiences with D&D and other TTRPGs to account for and potentially eliminate confounding variables. Gathering more diverse feedback would allow for a clearer understanding of user expectations and engagement patterns – especially considering the variety of player approaches and backgrounds in a typical D&D setting.

8.5.3 Smoothing Scenario Transitions

A common piece of feedback that we received from players is that, often the transition between scenarios can feel abrupt or interrupts follow up plans that players have, such as tending to wounded after a battle. Since the system only provide a fix window between detecting scenario resolution and scenario transition, we would like to investigate alternative methods for transitioning scenarios that leaves users more satisfied.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Nishank Gite, Eesha Pamula, Arnav Devineni, and Siddhant Rao. 2024. Enhancing structured narrative generation in language models: A fine-tuning approach utilizing short stories. *University of California, Berkeley*.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6379–6393, Online. Association for Computational Linguistics.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions.

HuggingFace. Bert model documentation.

HuggingFace. 2023. Qwen-2.5-72b model card. Accessed: 2023-12-12.

Ylber Januzaj and Artan Luma. 2022. Cosine similarity – a computing approach to match similarity between higher education programs and job market demands based on maximum number of common words. *International Journal of Emerging Technologies in Learning (iJET)*, 17:258–268.

Cheonsu Jeong. 2024. Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b. *Journal of Intelligence and Information Systems*, 30(1):93–120.

Shivani Kapania, Ruiyi Wang, Toby Jia-Jun Li, Tianshi Li, and Hong Shen. 2024. "i'm categorizing llm as a productivity tool": Examining ethics of llm use in hci research practices.

Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. 2016. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6.

LangChain. a. Inmemoryvectorstore.

LangChain. b. Langchain.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

OpenAI. a. Hello gpt-4o.

OpenAI. b. Openai o1.

Colby Phillips, Daniel Johnson, Peta Wyeth, Leanne Hides, and Mariusz Klarkowski. 2016. The validation of the game user experience satisfaction scale (guess). *Journal of Usability Studies*, 12(1):47–71.

Jonathan Rosen. 2023. Authors file lawsuit against openai, claiming copyright infringement over chatgpt. *The New York Times*.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Nisha Simon and Christian Muise. 2024. Tattletale: Storytelling with planning and large language models. *Queen's University, School of Computing*. 20nis@queensu.ca, christian.muise@queensu.ca.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 887–890. ACM.

Yuqian Sun, Hanyi Wang, Pok Man Chan, Morteza Tabibi, Yan Zhang, Huan Lu, Yuheng Chen, Chang Hee Lee, and Ali Asadipour. 2023. Fictional worlds, real connections: Developing community storytelling social chatbots through llms.

Phoebe J. Wang and Max Kreminski. 2024. Guiding and diversifying llm-based story generation via answer set programming.

Zhihua Wen, Zhiliang Tian, Wei Wu, Yuxin Yang, Yanqi Shi, Zhen Huang, and Dongsheng Li. 2023. Grove: A retrieval-augmented complex story generation framework with a forest of evidence.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023. The next chapter: A study of large language models in storytelling.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2024. The next chapter: A study of large language models in storytelling. *School of Computing and Information Systems, The University of Melbourne*. Zhuohanx@student.unimelb.edu.au, {t.cohn, laujh}@unimelb.edu.au.

A Appendix

Question	Expected Response
I am captivated by the game's story from the beginning	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I enjoy the fantasy or story provided by the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I feel detached from the outside world while playing the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I do not care to check events that are happening in the real world during the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I think the game is fun.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I feel bored while playing the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I feel the game allows me to be imaginative.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I feel creative while playing the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I am very focused on my own performance while playing the game.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I find the game social interactions believable.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I would like to play this game with other players.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I enjoy the game's dialog.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)
I think the game's is articulate.	1 - 7 | (1 = Strongly Disagree; 7 = Strongly Agree)

Table 3: GUESS-18 modified survey used.

Player Satisfaction Survey Results				
Category	Q	Mean	Mode	STD
Narrative	1	5.13	5	1.5
	2	5.6	6	1.35
Player Engrossment	3	4.27	6	1.87
	4	4.23	5	1.83
Enjoyment	5	5.67	6	0.816
	6*	2.53	1	1.85
Creative Freedom	7	5.87	7	1.25
	8	5.4	7	1.64
Personal Gratification	9	4.8	5	1.78
Social Connectivity	10	5.07	6	1.58
	11	5.67	7	1.40
Textual Aesthetics	12	5.07	6	1.49
	13	5.27	5	1.49

Table 4: User Satisfaction Scores for OpenAI backed chatbot. Question 6 asks if plays felt bored during play, and is the only question where a lower score is better.

Player Satisfaction by Category	
Category	Average Score
Narratives	5.37
Player Engrossment	4.27
Enjoyment	5.57
Creative Freedom	**5.63**
Personal Gratification	4.8
Social Connectivity	5.37
Textual Aesthetics	5.17

Table 5: Average user satisfaction score by Category

Model Satisfaction Survey Results				
Category	Q	GPT 4o	Qwen 14B	Qwen 7B
Narrative	1	**6.33**	5.33	5
	2	**5.67**	5.33	**5.67**
Player Engrossment	3	**5**	3	3
	4	**4.3**	4	2.33
Enjoyment	5	**5.33**	5	5
	6*	4.33	**4**	**4**
Creative Freedom	7	5	**6.33**	6
	8	6	**6.3**	6
Personal Gratification	9	**5**	4	4.33
Social Connectivity	10	**5.67**	4.33	5
	11	**6.33**	3.67	3.33
Textual Aesthetics	12	**6**	3	2.33
	13	**6.33**	3.67	3.33

Table 6: Team members satisfaction results based on model.

Scenario Number	**Cosine Similarity Score**
1	0.7085
2	0.7827
3	0.8199
4	0.8671
5	0.7684
6	0.6830
7	0.7094
8	0.7696
9	0.8385
10	0.5850
11	0.7141
12	0.7140
13	0.6932
14	0.7117
15	0.5767
16	0.8566
17	0.7776
18	0.7686

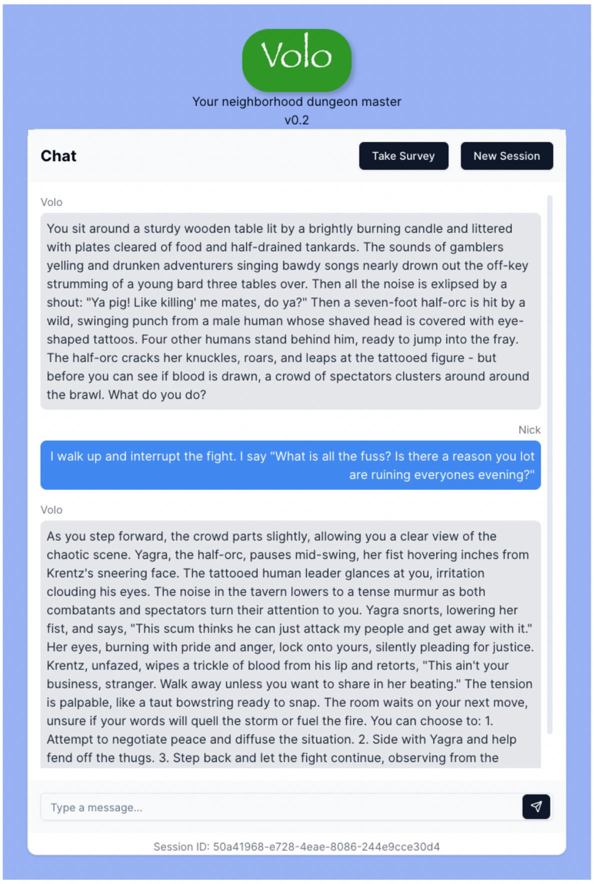
Table 7: Cosine Similarity Scores between the generated events and the corresponding passage in the module "Deepwater - Dragon Heist" using '*all-mpnet-base-v2*' sentence transformer.

Story Progression Metrics								
Scenario Level	Average per Turn		Entire Chat			Average per Turn		
	BLEU1	BLEU2	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
1	0.39	0.14	0.19	1.07e-2	3.44e-2	1.18e-3	3.87e-4	1.18e-3
2	0.43	0.17	0.20	1.54e-2	3.60e-2	2.41e-3	1.08e-3	2.41e-3
3	0.46	0.22	0.14	1.54e-2	3.30e-2	4.63e-3	2.19e-3	4.64e-3
4	0.41	0.16	0.20	1.70e-2	3.62e-2	2.51e-3	1.18e-3	2.51e-3

Table 8: Quantitative metrics averaged across all players that played to specific scenario number. Average per Turn indicates the average for each score when measured comparing the generated chat for that message and the entirety of the text content chapter 1 of the adventure module. Entire Chat rouge means that the contents of the entire chat history is used to score against the entirety of chapter 1. The Rouge per turn average is calculated differently than the BLEU score average. Instead, the Rouge score is calculated for each chat session and divided by the number of generated messages. Then these averages are averaged again across all sessions. Stop Words were removed from both chapter 1 text and reference material before scoring.

Story Progression Statistics		
Scenario Number	Num Players Reached	Avg Chat Hist Length
1	2	32
2	8	20
3	2	8
4	3	22.5

Table 9: Break down of how many players finished their play session and in which acts and how many turns on average it took them to reach there.



Volo

You sit around a sturdy wooden table lit by a brightly burning candle and littered with plates cleared of food and half-drained tankards. The sounds of gamblers yelling and drunken adventurers singing bawdy songs nearly drown out the off-key strumming of a young bard three tables over. Then all the noise is exlipsed by a shout: "Ya pig! Like killing' me mates, do ya?" Then a seven-foot half-orc is hit by a wild, swinging punch from a male human whose shaved head is covered with eye-shaped tattoos. Four other humans stand behind him, ready to jump into the fray. The half-orc cracks her knuckles, roars, and leaps at the tattooed figure - but before you can see if blood is drawn, a crowd of spectators clusters around around the brawl. What do you do?

Nick

I walk up and interrupt the fight. I say "What is all the fuss? Is there a reason you lot are ruining everyones evening?"

Volo

As you step forward, the crowd parts slightly, allowing you a clear view of the chaotic scene. Yagra, the half-orc, pauses mid-swing, her fist hovering inches from Krentz's sneering face. The tattooed human leader glances at you, irritation clouding his eyes. The noise in the tavern lowers to a tense murmur as both combatants and spectators turn their attention to you. Yagra snorts, lowering her fist, and says, "This scum thinks he can just attack my people and get away with it." Her eyes, burning with pride and anger, lock onto yours, silently pleading for justice. Krentz, unfazed, wipes a trickle of blood from his lip and retorts, "This ain't your business, stranger. Walk away unless you want to share in her beating." The tension is palpable, like a taut bowstring ready to snap. The room waits on your next move, unsure if your words will quell the storm or fuel the fire. You can choose to: 1. Attempt to negotiate peace and diffuse the situation. 2. Side with Yagra and help fend off the thugs. 3. Step back and let the fight continue, observing from the

Session ID: 50a41968-e728-4eae-8086-244e9cce30d4

Figure 2: Volo chatbot interface