NLP-Driven Dungeon Master for D&D 5e

Overview

In this project we aimed to create an AI Dungeon Master for D&D 5e. We used the meta-llama/Llama-3.2-3B-Instruct model to create a fantasy world, kingdoms, towns, NPCs and gameplay interactions. We decided to not continue developing our previous model with gpt2-large as the responses were very poor and therefore opted for Llama-3.2-3B model.

To develop the dungeon master we used the following files:

World creation. ipynb

This notebook is focused on world-building and provides the necessary tools to create a setting for the project. It contains:

1. Fantasy World Generation:
    - We use the meta-llama/Llama-3.2-3B-Instruct model from hugging face to generate detailed descriptions of the fantasy world.
2. Kingdom and Town Creation:
    - Creates kingdoms, towns and NPCs within the generated world. Each location also has its own unique characteristics like resources and cultural elements.
3. Prompt Engineering:
    - We use detailed prompts for the language model to enhance creativity and generate more engaging outputs.
4. Saving the world:
    - We store the world generated and the elements in it in the form of a dictionary. The dictionary is saved as a Json file for later use.

Start. Ipynb

Functionality

This notebook implements the gameplay mechanics for the project. It builds on the outputs from world to enable a dynamic gaming experience. Key features:

1. World Loading and Management:
    - Loads pre-generated worlds saved as JSON files before from the previous notebook. This allows continuity and integration between world-building and gameplay.
2. Gameplay Logic:

- It implements the main action loop for the game. Actions can be like (e.g., exploring, interacting with NPCs, engaging in battles), and the AI responds accordingly.

3. Save and Load files:
    - It supports saving and loading game states so that players can resume their progress at any point.
4. Starting Point:
    - We store the starting point generated by the model along with the world information in a dictionary named the game state in a Json file for later use.

helper2.py

This file is the helper module for the AI Dungeon Master which contains all the essential functions and tools to manage the game state, load world data, and facilitate gameplay interactions using the model.

1. Model Initialization

- Model Setup:
    - It uses the hugging face transformers library to set up a text-generation pipeline.
    - Model: meta-llama/Llama-3.2-3B-Instruct
    - Optimized for text generation with device settings of (torch. bfloat16 for efficient computation on GPU).

2. World Management

- load_world(filename):
    - Loads a JSON file containing the pre-generated fantasy world before.
    - This file contains information about kingdoms, towns, NPCs, and the starting scenario.
- get_game_state(inventory={}):
    - Constructs and returns the initial game state, including:
        - World Description: Description of the fantasy world.
        - Kingdom and Town Details: Descriptions of the starting kingdom and town.
        - Character Description: Details about the main NPC in the starting town.

        - Player Inventory: Tracks items the player has collected.
- print_game_state():
    - It iterates through the world data and prints the view of kingdoms, towns, and NPCs.
    - The function was not called in the main file as the response time increased for some reason when invoked.

3. Gameplay Interaction

- run_action(message, history, game_state):
    - Processes player commands and generates the next step in the game based on the current game state and history.
    - Core Features:
        - It utilizes a system prompt to direct the language model on generating context related responses.
        - Incorporates:
            - The world's description.
            - The current kingdom, town, and character details.
        - Tracks interaction history to maintain context across multiple player actions.
    - Output:
        - Generates between 1-3 sentences in response to player input, maintaining consistency and immersion.

game.py

This file is the main driver for the AI Dungeon Master. It integrates all functionalities such as inventory management, story generation, and user interaction. Additionally, it creates a web-based interface for gameplay using Flask.

Functionality

1. Model Initialization

- Pipeline Setup:
    - Configures the Hugging Face Transformers pipeline with the meta-llama/Llama-3.2-3B-Instruct model for text generation.
- Torch and Hugging Face Authentication:

    - Manages GPU memory with torch.cuda.empty_cache().
    - Authenticates with Hugging Face Hub for accessing the model.

2. Inventory Management

- System Prompt for Inventory Updates:
    - Analyse recent story outputs and detects changes to the player's inventory.
    - Adds or removes items based on player actions.
- detect_inventory_changes(game_state, output):
    - Extracts inventory changes (items gained or lost) based on the generated story.
    - Parses the model's output to extract valid JSON containing inventory updates.
- update_inventory(inventory, item_updates):
    - Player's inventory is updated based on detected changes.
    - It also deals with edge cases like items with negative quantities and removes them from the player inventory.

3. Gameplay Interaction

- System Prompt for Game Actions:
    - Guided the language model to generate short and relevant responses to player actions.
    - Enforces game rules, such as preventing players from using items not in their inventory.
- run_action(message, history, game_state):
    - Processes player input, incorporates past interactions (history), and generates the next narrative step.
    - Checks for logical consistency by using inventory information
- main_loop(message, history=None):
    - Combines action generation and inventory updates into a single loop.
    - Maintains player interaction history and updates the game state dynamically.

4. Web-Based Interface

- Flask Integration:
    - Provides a web interface for gameplay using Flask.
    - Routes:

        - "/": Renders the main game interface.
        - Supports both GET and POST requests to handle user input and return AI responses.
- Gameplay Workflow:
    - The player interacts with the AI through a web form.
    - The AI responds based on the player's actions and the current game state.

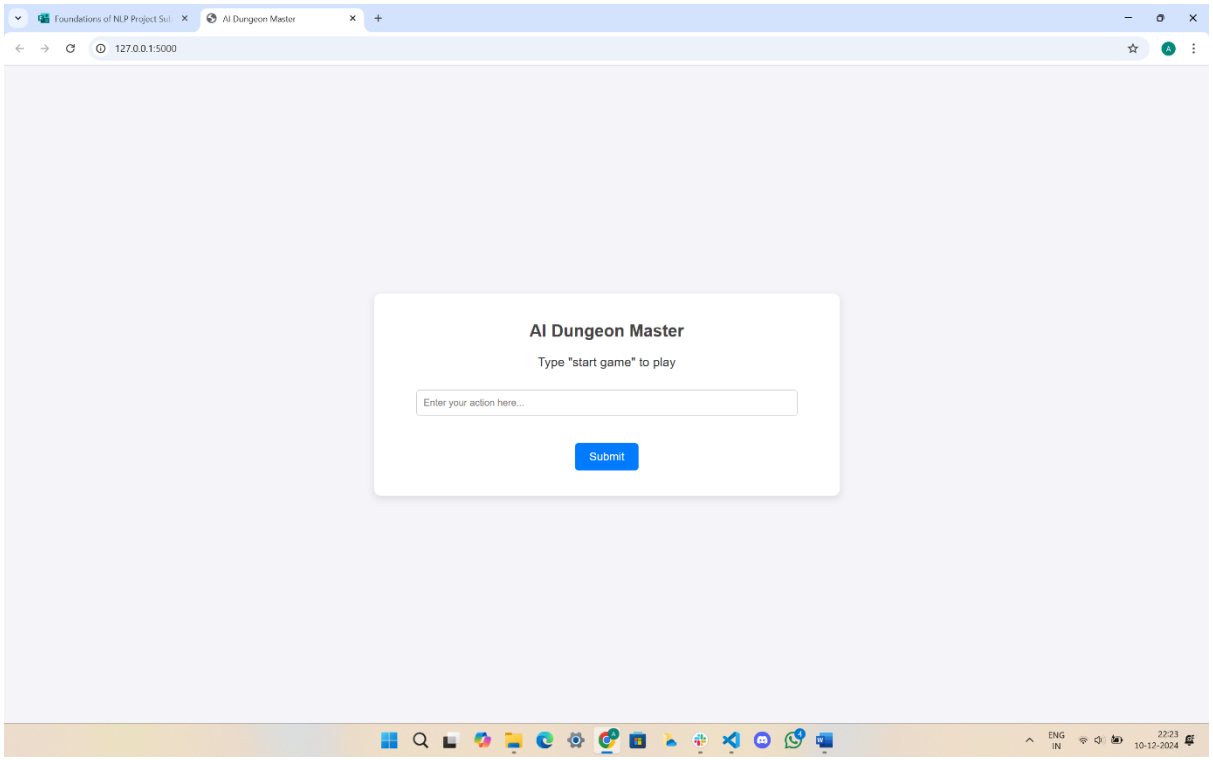
5. World and Game State Integration

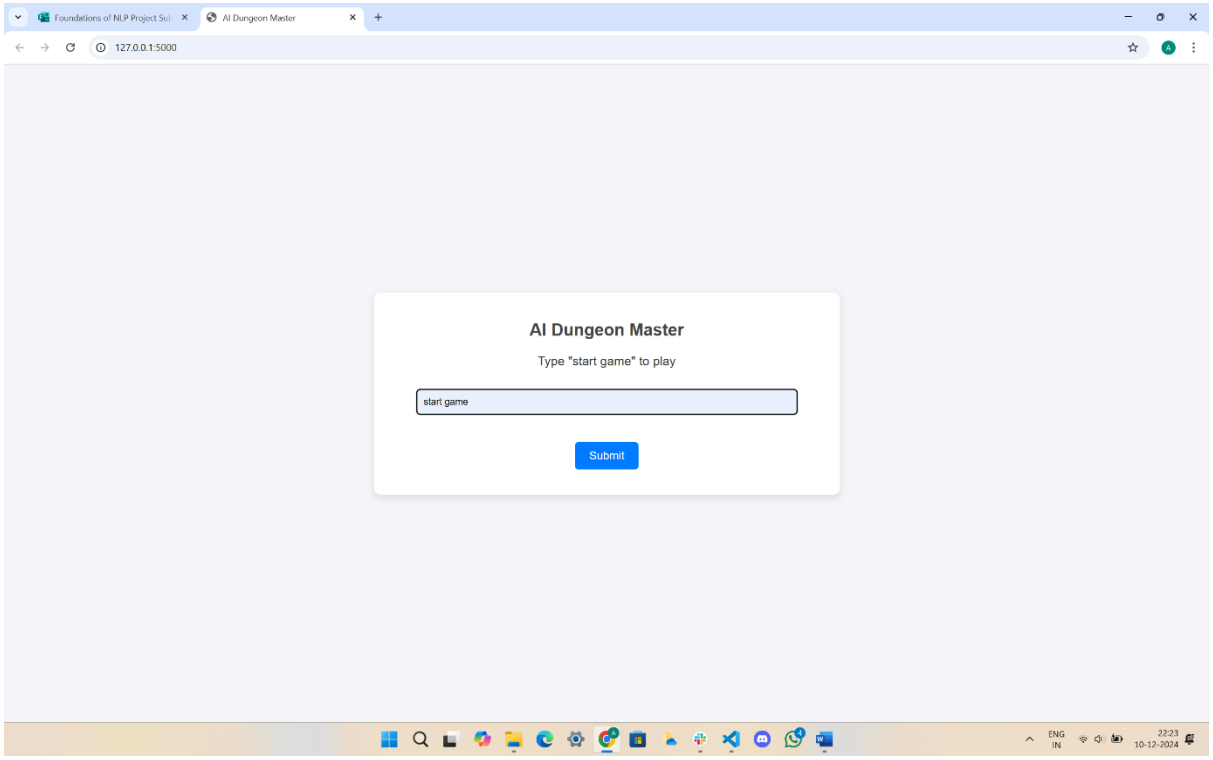
- Imports functionalities from helper2.py, such as:
    - get_game_state(): Loads the game world and initializes the player's inventory.
    - print_game_state(): Outputs the structure of the game world for debugging or reference.

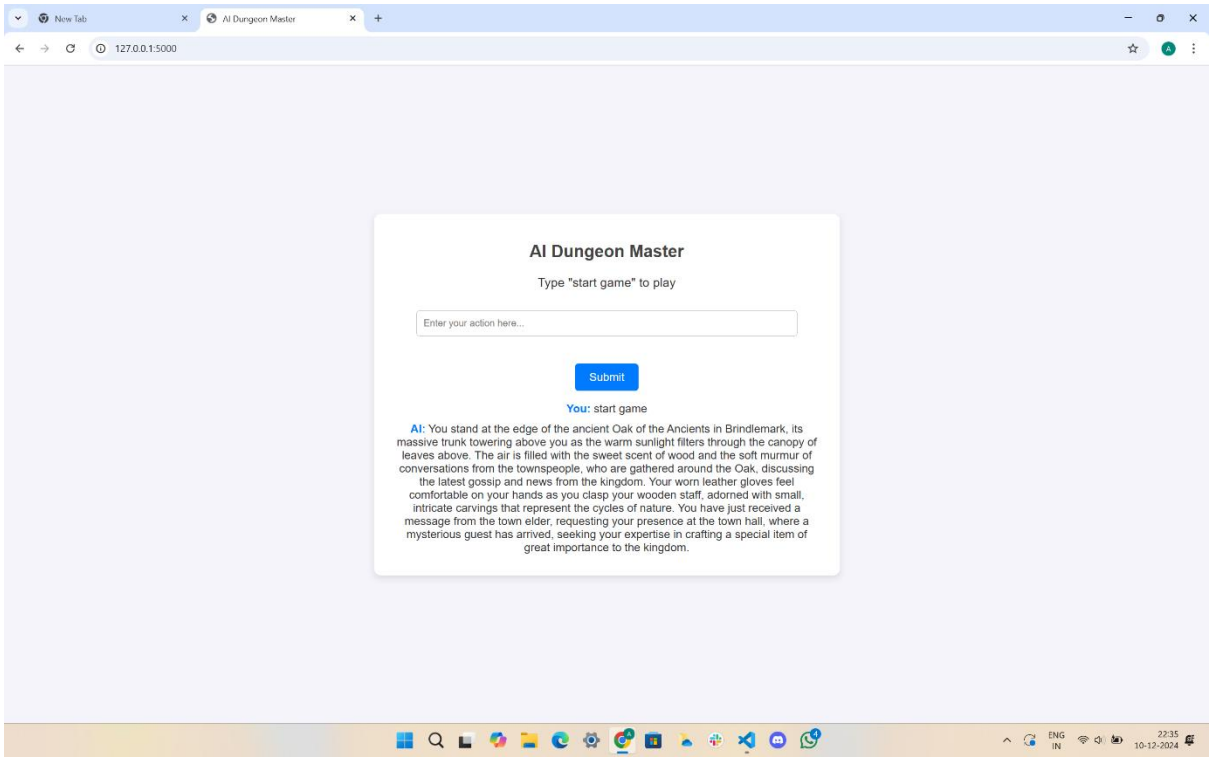
index.html

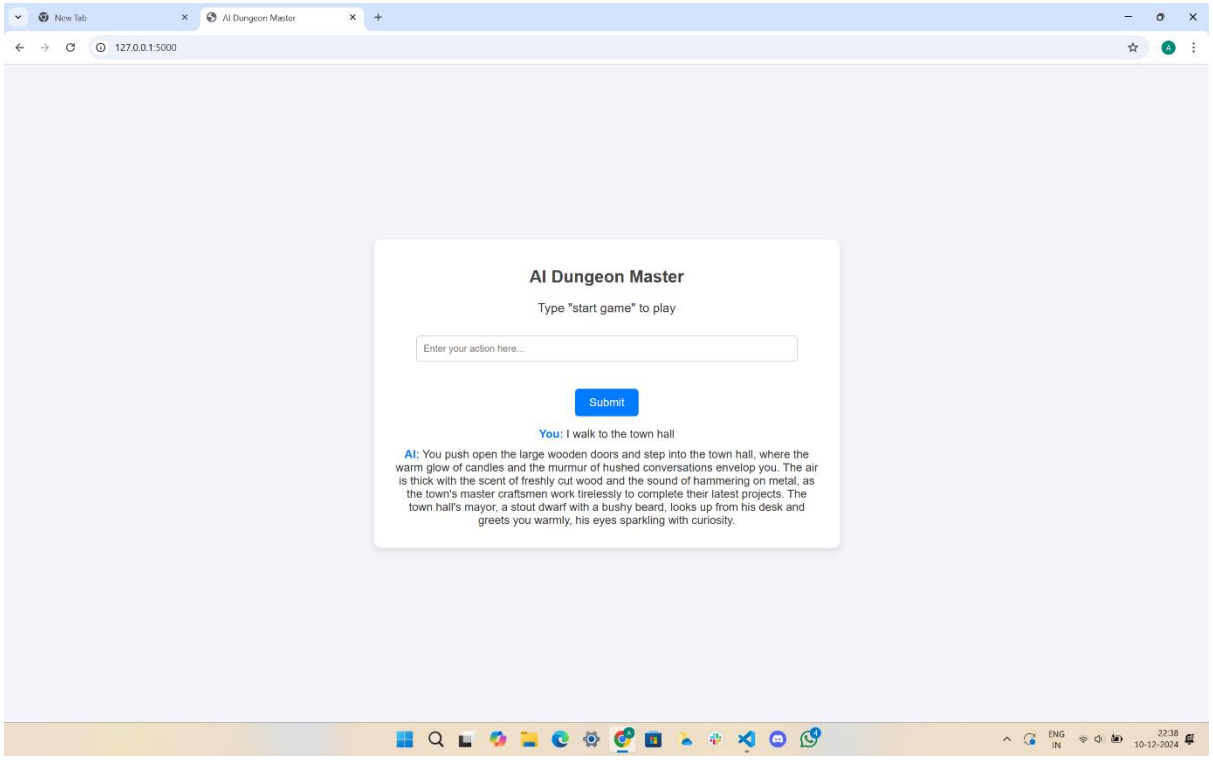
This is the user interface for the web-based AI dungeon master. It provides an interactive platform for players to input their actions and receive dynamic responses from the AI Dungeon Master.

Gameplay


Foundations of NLP Project Sul
AI Dungeon Master
127.0.0.1:5000
AI Dungeon Master
Type "start game" to play
Enter your action here...
Submit
ENG
IN
22:23
10-12-2024


Foundations of NLP Project Sul
AI Dungeon Master
127.0.0.1:5000
AI Dungeon Master
Type "start game" to play
start game
Submit
ENG
IN
22:23
10-12-2024


New Tab
AI Dungeon Master
127.0.0.1:5000
AI Dungeon Master
Type "start game" to play
Enter your action here...
Submit
You: start game
AI: You stand at the edge of the ancient Oak of the Ancients in Brindlemark, its massive trunk towering above you as the warm sunlight filters through the canopy of leaves above. The air is filled with the sweet scent of wood and the soft murmur of conversations from the townspeople, who are gathered around the Oak, discussing the latest gossip and news from the kingdom. Your worn leather gloves feel comfortable on your hands as you clasp your wooden staff, adorned with small, intricate carvings that represent the cycles of nature. You have just received a message from the town elder, requesting your presence at the town hall, where a mysterious guest has arrived, seeking your expertise in crafting a special item of great importance to the kingdom.
ENG
IN
22:35
10-12-2024


New Tab
AI Dungeon Master
127.0.0.1:5000
AI Dungeon Master
Type "start game" to play
Enter your action here...
Submit
You: I walk to the town hall
AI: You push open the large wooden doors and step into the town hall, where the warm glow of candles and the murmur of hushed conversations envelop you. The air is thick with the scent of freshly cut wood and the sound of hammering on metal, as the town's master craftsmen work tirelessly to complete their latest projects. The town hall's mayor, a stout dwarf with a bushy beard, looks up from his desk and greets you warmly, his eyes sparkling with curiosity.
ENG
IN
22:38
10-12-2024

Challenges Faced

Some of the challenges we faced while making the AI Dungeon Master are-

- The model due to its size takes a bit of time to generate responses as we have limited resources
- The initial model made using gpt-2 large gave very out of context responses despite our efforts in refining the prompts which led to us changing our model.
- We tried showing the initial game state to the user when the user starts the game but this led to increase in time for the responses so we did not implement it.