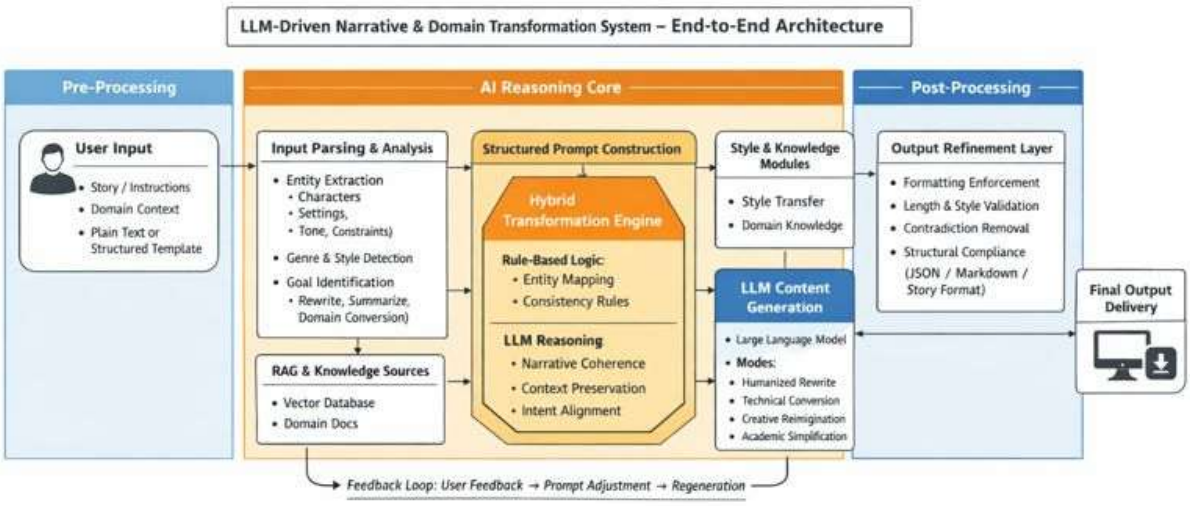
Approach diagram:



System design - How it works:

Story Generator is AI system powered by LLM generates you the story based on the inputs given by user and using series of prompts using Langchain, the re-imagined story is generated.

I used single leader, **DAG** (Directed Augmented Graph) style orchestrated pipelinep

The AI system works on the input provided by the user that user has to enter the story and the desired universe needed to make it a fantasy.

This system checks if the story is public domain or not with help of API of the LLM through web

After verifying the story, it proceeds to the next step.

The following steps are to generate a re-imagined story

- **Extraction of the original story**
- **Making a different set of rules to setup a new universe**
- **Mapping the characters of the original story to the new characters of re-imagined story**
- **Transforming the plot**
- **Final story generator**
- **Consistency checker**

Extraction of the original story

The system makes the overview of the story and understands the essence of the story by taking in account of what is story based upon, how the characters are mentioned, what are the places mentioned, what scenario is it based on, what plot is changing the character and so on.

Making a different set of rules to setup a new universe

This module builds up a set of rules for the new universe mentioned by the user. It fetches the basic idea about how the universe works nah the timeline of the universe from references which is obtained from the LLM.

Mapping the characters of the original story to the new characters of re-imagined story

The extracted characters from the original story with their properties mentioned in the story are mapped to the created new characters of the re-imagined story, and it is taken in account.

Transforming the plot

The plot is designed based on the set of rules made in the module and the plot is generated based on the essence of the original story but for the universe which is mentioned in the input.

Final story generator

In this module, the system generates the story based on the data's which are taken in the account. The essence, the characters, rules set for the universe and finally, the plot.

Consistency checker

This module keeps the track on the modules which are responsible for each module that is for generating the re-imagined story. It also keeps a check on the prompts to ensure the difference from the original story

Alternatives

1. **Fully prompt-based design**

   This method works on making a prompt as a whole single prompt. The drawback which lead me not to choose this it increases the complexity of the debugging and it is a weaker method that cannot withstand longer prompts, it crashes and it also creates the plot of the story that doesn't seems have essence of the story.

2. **Structured multi-stage pipeline (Chosen)**

   I chose this method because I can keep track of what is happen and it is easy to debug and it is also provides a stabler one compared to other alternatives. The modules can be updates easily for further improvisions**.**

3. **Few-shot prompting**
   This method is based on the providing multiple references to the model. It can be efficient when the quality of references is good and no. of references are larger, and it makes errors when shifting the universe because of the multiple references provided in the dataset. This leads to lack of the essence of the story or the plot of the story.

Challenges & Mitigations

Maintaining the essence of the story. Sometimes, if the story and universe don't match the characters fell apart. It sounded like a irrelevant re-imagined story

For this, I broke the sequence by mentioning the details to compare the story generation and the given universe based on the information fetched from LLM and map it the outline of the new story and to follow the universe rules. So, I built consistency checker module.

The tone of the story gets shifted in middle. So, I made consistency checker to pre-process the tone in a loop until it is correct

The facts were getting fabricated in some parts of the story. The RAG which I implemented made sure to check those and correct it at the time of final output

Improvements

It is now a system without no front-end user interface

What can be added in

Terms of Production line is

- It can be made as a website for our particular product
- We can make user accounts to store their generated stories

Terms of Technical scaling

- We can integrate vector database such as pinecone/ FAISS to ensure the stability of the model
- We can fine tune the models based on genres for better outputs
- We can make templates universes
- A chatbot that gives suggestion on the story and universe selection

Additional features which can be added are

- Collaboration mode
- Plagiarism checker
- Tone slider
- Export formats
- Automatic outline generator