Short Story Long : An Automated Story Generating Framework with Character Profile Creation and Social Interaction Dialogues

Shristi Suman
University of California, Davis

Rohith Raj Srinivasan
University of California, Davis

Shreyas Madhav Ambattur Vijayanand
University of California, Davis

Abstract

Automated story creation and plot generation techniques for stories/movies/video games have been extensively to assist in automating the creative processes involved in creating storylines while retaining the original creative influence of the authors has established itself as a prominent field of research over the past decade. Most story generation systems work upon a set of predefined inputs from the author that are mandatory for their functioning which may be restrictive to the creative potential of authors, considering how diverse the human storymaking process can be. Our proposed project Short Story Long (ShortSL) aims to provide a complete integrated framework that can assist any author regardless of their thinking process in most idea generation/brainstorming/creation processes involved for creating a game's storyline, characters, appearances, in-game dialogues, etc. This helps accommodate authors of different thinking characteristics on a single automated platform to help in story generation, character profile creation, and dialogue generation. The story and dialogue generation processes of our ShortSL backend system(GPT-3) was evaluated against BERT, GPT-2, and RoBERTa. The results achieved on this comparative analysis have been promising.

Keywords: *Multi-Modal Storytelling, Story Generation, Character Creation, Dialogue Generation, Image Generation, Natural Language Processing, Text-to-Image Synthesis.*

1 Introduction

Computer-based story generation techniques focus on creating event sequences, prompting scenes, character action and text-based narratives to come up with a coherent storyline or plot that can be utilized in the creation of novels, video games, movies etc. Story generation systems can be categorized under multiple facets based on different criteria. Non-learning methods of generation depend heavily upon existing knowledge structures that have been hand-coded to facilitate the building of reliable knowledge containers. While these methods can be used to provide stories that are considered "legal" based upon predefined rules and human knowledge, most such symbolic domain models or case-based reasoning systems need to be engineered effectively to understand whether its usefulness is driven by proper knowledge engineering practices or by the algorithm itself. The stories generated will also be entirely restricted to the knowledge provided and cannot accommodate all creative decisions of the authors.

Learning-based methods of story generation primarily rely on natural language processing and deep learning to understand the user's expectations from text prompts and formulate a coherent story with good flow based upon previously learned knowledge from actual human-written stories. When dealing with such automated assistive technologies that aim to help creative thinking, it is mostly preferred for it to be a mixed approach where the human creator possesses complete authority over the creative content by providing one or more inputs to the system on how the story should evolve and which characters should attain the spotlight. An ideal system should also provide the creators to make changes to the generated stories in such a way that re-generation results should make the changes while keeping the main idea intact. Hence, the author should be in control of what components of the story (intro, genre, climax, characters, personality traits, etc.) must be decided by the AI and what components should be prompted by the author. It is important to note that different human authors have different approaches to formulating a story (some start from climax, some define their characters first etc.) so it will be extremely efficient if the automated system can accommodate these creative differences.

Traditionally, game plots were created through manual scripts, which involved a lot of brainstorming, writing, and further editing of the events, the storyline and the characters involved in the game. However, Artificial Intelligence (AI) has dramatically impacted this process by automating the generation part to a great extent. Using AI algorithms, game developers can generate events, storylines, and characters based on particular input and parameters provided by the user. These algorithms are further trained on a large dataset of existing plots, allowing them to learn present plot structure and generate diverse and unique plots saving time and effort required, which would not have been possible otherwise. Genre-based story generation generates stories belonging to particular genre-science fiction, romance, fantasy, or mystery. Each genre will have its distinct features and conventions. The main goal is to create an engaging narrative that compels the player to continue playing and satisfies the expectation associated with the genre.

Character-based story generation generates stories around multidimensional characters that drive the story's narrative. In this type of storytelling, the goal is not simply to advance a plot but to focus on the personal experiences and journeys of the involved characters. Generally, each character will have certain personality traits, motivations, goals, and a backstory. Scene-based story generation refers to creating a narrative by generating a series of scenes that capture the key events and actions. The scenes are usually brief descriptions of events or actions, and serve as building blocks that come together to create a complete story.

The aforementioned story-generation techniques can be achieved using AI, which is widely popular in the research community. Several AI techniques, such as machine learning, natural language processing, and deep learning algorithms, are prevalent in this domain. For a genre-based generation, the AI system uses large datasets of texts belonging to a particular genre, such as romance, science fiction, or horror, to generate new stories. Moreover, for a character-based generation, a large corpus of story examples and character descriptions is used to train an AI system to learn the characters' attributes and the relationship between characters. Furthermore, for a scene-based generation, the program uses machine learning algorithms and a large corpus of text to generate coherent scenes that follow the conventions of storytelling. Generative Adversarial Network (GAN) is a deep learning model used for generative modeling, which involves creating new data samples that resemble some existing data. GANs have been used for various applications such as image generation, style transfer, and data augmentation. Moreover, it has been widely used in the game industry to generate impressive visual character profiles.

Apart from the main story generation tasks, AI systems have also gained their role within the storytelling narratives, especially in in-game dialogue generation for social interactions between two or more characters in the game. Authors can input what situation and scene the social interactions are set in along with the emotions or endpoints that are to be achieved through it, and the learning-based systems should generate utterance pairs to replicate realistic social expressions and dialogue. The expected linguistic form is governed by derived parameters set that can keep in mind the emotional attachments and affinities among the story characters.

Our project answers the below questions:

- Providing maximal creative control to the author and accommodating creative differences by creating a system that can adapt to input differences instead of one predefined input?
- Generating Character profiles and backstories based upon minimal input from authors and the main plot?
- Can we accommodate dynamic changes proposed by the author as the story is generated and regenerated with changes while maintaining logic, flow, and coherence?
- Giving Authors creative control over the character visuals before supplying them to the animator?
- Helping authors model dynamic social interactions between two or more game characters based upon one or more prompted author expectations (personality traits, in-game location, genre, outcome etc.) with option to spontaneously generate in-game dialogues, along with the ability to regenerate portions of the dialogue that the author wants edited without losing the original crux of the dialogue?

2 RELATED WORK

Automated story creation [1] and plot generation techniques for video games have been extensively explored in the past to assist in automating the creative processes involved in video game storylines while retaining the original creative influence of the authors [2]. In most past such systems, authors could provide the system with prompts or critical elements of the story that they have in mind, and the system uses this initial information to develop a storyline for the game. The intuitive input that is received in most cases are scene fragments, turn points or climaxes of the stories that the authors have visualized in their mind, and the AI system connects the dots by intertwining all of these scenes, characters, and key elements into a coherent playable gameplay plot.

Wide Ruled [3] is one of the prominent interfaces established in this space for crafting and writing generative stories. The interface of Wide Ruled is designed with the non-technical author in mind, featuring familiar interface elements, clear narrative terms, and step-by-step guidance through the creation of even complex plans and actions. Graph-based approaches have also been proposed for plot outlining. A layered hierarchical graph is a tool for outlining a game's plot. The graph generation system is unique in the sense that it portrays the player's journey through the game world using a series of rules [4].

Character-based plot generation has also gained much traction in the Game AI world, where the creator supplies the systems with protagonists, NPCs, and other characters in a particular environment that they would like to include in the plot. The AI uses this information to generate the storyline. In character-centric systems, each story aspect is closely tied to a particular character and corresponding environment. This approach explicitly incorporates character information and the relationships between the plot and character to enhance clarity and coherence in the generated story [5].

One direction worth exploring is the visual generation of video game characters based on physical characteristics and personality trait descriptions [6] and vice versa [6].

There have been several advancements in controlling the generation of stories. The genre of the story is one significant element for generating personalized stories. Recently, there has been a growing interest in using AI and machine learning techniques to generate genre-based stories. Cho et al. [7] have successfully leveraged contrastive learning and developed a supervised contrastive learning model for genre-controllable story generation. Liu et al.l [8] have proposed a character-centric neural storytelling model to address the models that only consider linguistic patterns while generating stories. This model generates character-centric stories by considering the character, its corresponding attributes, and the story genre while generating stories, and it has outperformed several traditional generation models.

Moreover, the technique of generating stories based on climax has existed since time immemorial, and there have been several successful attempts to use AI for the same. For instance, Peng et al. [9] have developed a Recurrent Neural Network (RNN) based framework to enable controllable story generation. They have primarily focussed on the storyline of stories and the ending valence of stories as the control elements. Their framework successfully achieved the desired control for generating personalized and coherent stories.

Character creation for games requires much proficiency; building them from scratch is a time-consuming process. However, using Generative Adversarial Network(GAN) provides an alternative that allows for the creation of custom characters into existence without any professional skill. GAN uses a combination of a deep learning network and a competitive training algorithm and has been used widely to produce realistic images. Anime character generation is one of the most major work done, and Yanghua et al. [10] has approached this problem by collecting a clean dataset and using an empirical application of DRAGAN.

Creating interactive and dynamic stories is one of the crucial aspects of modern game design. Generative grammar has been used to create written dialogues, and sentiment analysis has been employed to ensure semantic quality [11]. Intentional Dialogue Line is one of the other proposed works where multiple dialogue variants are generated and refined using Machine Learning to integrate dialogue lines into existing content while still being unique and fitting for the character speaking them.

3 SHORTSL - METHODOLOGY

3.1 Adaptable Author Input Module

The system starts with a text multi-input system where the author can enter one or more attributes of the story they want to generate. The fields that we have created include genre, start of the story, character information and traits, intermediate scenes or incomplete event sequences, and climax. The author can fill these fields according to what they have already created, and the system will generate the story based on these attributes. There is no mandate for the author to fill one or more of these fields. This has been done to accommodate the different creative thinking methods of different authors(some may create their story starting from significant events, some define their characters first etc). The inputs provided by the author will be passed on to the story generation module that will use the author preferences to create the storyline.

3.2 Story Generator Module

Once the author inputs have been fed to the story generation module, we adopt a deep learning natural language processing module to come up with the storyline, according to author preferences. We will be employing the GPT model family for this purpose. Generative Pre-Trained Transformer(GPT) is an unsupervised generative language model which takes input as a sentence and generates an

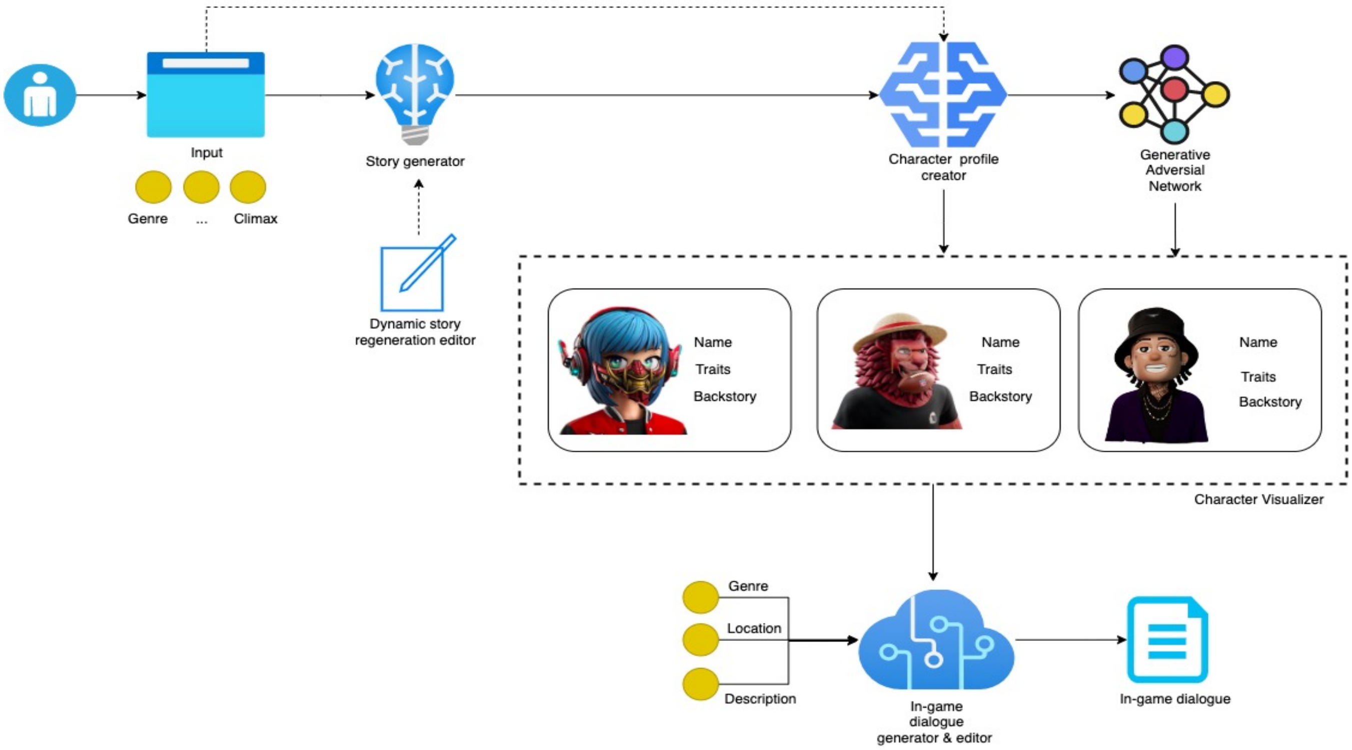


Figure 1: ShortSL System Architecture

appropriate response. GPT-2 and GPT-3 are different versions that are open source and are trained on massive amounts of text data, making it capable of generating the next sequence of text similar to input. For the scenario of story generation, GPT can be used to generate a preliminary concept or outline of a story. A prompt like the description of the character or a couple of sentences, can be given as input, and the model will generate text to continue the narrative and move ahead with the story. They can be further adjusted to produce more specific or focused output by training on a particular genre or style of writing. Hence, we will be using this approach to generate the initial story to be presented to the author.

3.3 Dynamic Story Regeneration Editor

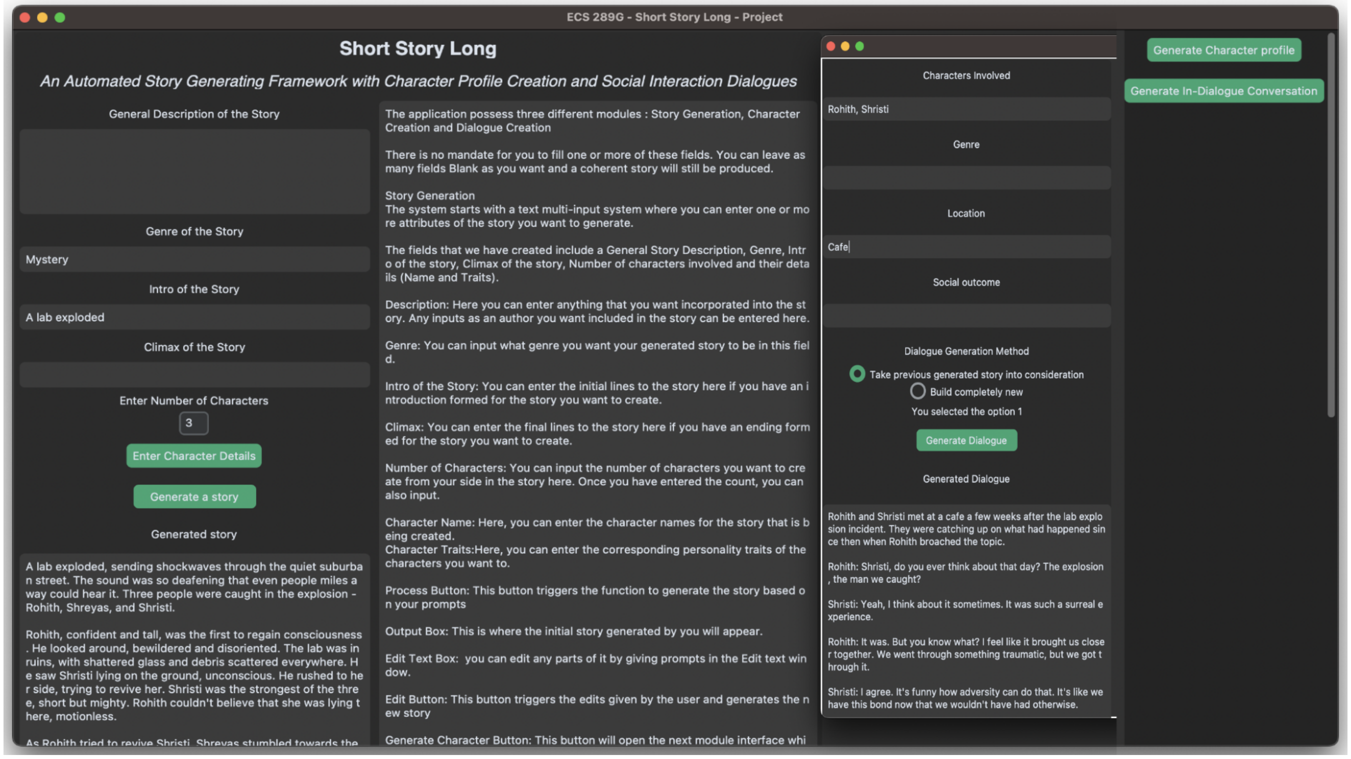
This module of the system succeeds the initial story generation process, is optional, and called according to the author's convenience. Once the initial story is generated according to the author's inputs, the author can choose to edit parts of the story or modify characters/scenes and request a regeneration. This regeneration will be fed as a consecutive query to the original request and the NLP model to make the changes while maintaining the main idea and crux of the story intact. The author can make as many edits to the story as required. The author can also choose to edit the initial inputs dynamically at any time in order to regenerate the storyline.

3.4 Character Profile Creator

Once the final story has been created and approved by the author, the system provides the author with the ability to generate character profiles. The profiles will consider the initial inputs provided. The traits or attributes not mentioned by the author will be generated automatically by language models based on the story. The character profile will display the Name, Traits (Strengths, Weaknesses, Appearances, etc.), and the backstory of the character (which will be generated based on the main story). The author can choose to edit these backstories or any other features of the character dynamically at any point in time. The system will also provide a virtual character appearance based on the description initially provided by the author, which will be powered by the Character Visualizer, explained in the next section.

3.5 Character Visualizer

The Character Visualizer aims to visualize the physical appearance of the characters based on author preferences and the main storyline that has been generated along with the character profile generated by the character profile creator module (discussed above). Having this module is very essential for our system since it provides the authors with the ability to get an overall idea of the characters involved in the system and provides the authors with the flexibility to further customize the characters in case they are not satisfied with it. Generative Adversarial Networks (GAN) have been widely used to synthesize new data akin to the data it has been trained on. So we intend to leverage GAN to generate virtual character appearance for our system and make it visually appealing for the authors. Generative Adversarial Network(GAN) is a class of machine learning framework used for generative modeling. Made of two components, generator creating synthetic data and discriminator checking if generated data is real or fake, GAN is used to create new data instances that resemble training data. The generator tries to fool the discriminator, and the discriminator tries to keep from being fooled and with continuous training the generator becomes better at creating realistic data and discriminator at identifying fake data. It is widely used in image and video generation but can also be used for story generation by training on large corpus of text data. DALL-E is a very powerful tool which uses natural language descriptions of the scene to generate relevant images. Since we have the character traits and the generated story, we can combine them and supply it to DALL-E to generate avatars of characters and embed it along with the profile of the character as shown in the system architecture diagram.



ECS 289G - Short Story Long - Project

Short Story Long

An Automated Story Generating Framework with Character Profile Creation and Social Interaction Dialogues

General Description of the Story

Genre of the Story: Mystery

Intro of the Story: A lab exploded

Climax of the Story

Enter Number of Characters: 3

Generated story

A lab exploded, sending shockwaves through the quiet suburban street. The sound was so deafening that even people miles away could hear it. Three people were caught in the explosion - Rohith, Shreyas, and Shristi.

Rohith, confident and tall, was the first to regain consciousness. He looked around, bewildered and disoriented. The lab was in ruins, with shattered glass and debris scattered everywhere. He saw Shristi lying on the ground, unconscious. He rushed to her side, trying to revive her. Shristi was the strongest of the three, short but mighty. Rohith couldn't believe that she was lying there, motionless.

As Rohith tried to revive Shristi, Shreyas stumbled towards the

The application possess three different modules : Story Generation, Character Creation and Dialogue Creation

There is no mandate for you to fill one or more of these fields. You can leave as many fields Blank as you want and a coherent story will still be produced.

Story Generation
The system starts with a text multi-input system where you can enter one or more attributes of the story you want to generate.

The fields that we have created include a General Story Description, Genre, Intro of the story, Climax of the story, Number of characters involved and their details (Name and Traits).

Description: Here you can enter anything that you want incorporated into the story. Any inputs as an author you want included in the story can be entered here.

Genre: You can input what genre you want your generated story to be in this field.

Intro of the Story: You can enter the initial lines to the story here if you have an introduction formed for the story you want to create.

Climax: You can enter the final lines to the story here if you have an ending formed for the story you want to create.

Number of Characters: You can input the number of characters you want to create from your side in the story here. Once you have entered the count, you can also input.

Character Name: Here, you can enter the character names for the story that is being created.
Character Traits:Here, you can enter the corresponding personality traits of the characters you want to.

Process Button: This button triggers the function to generate the story based on your prompts

Output Box: This is where the initial story generated by you will appear.

Edit Text Box: you can edit any parts of it by giving prompts in the Edit text window.

Edit Button: This button triggers the edits given by the user and generates the new story

Generate Character Button: This button will open the next module interface whi

Characters Involved: Rohith, Shristi

Genre

Location: Cafe

Social outcome

Dialogue Generation Method
● Take previous generated story into consideration
○ Build completely new
You selected the option 1

Generated Dialogue

Rohith and Shristi met at a cafe a few weeks after the lab explosion incident. They were catching up on what had happened since then when Rohith broached the topic.

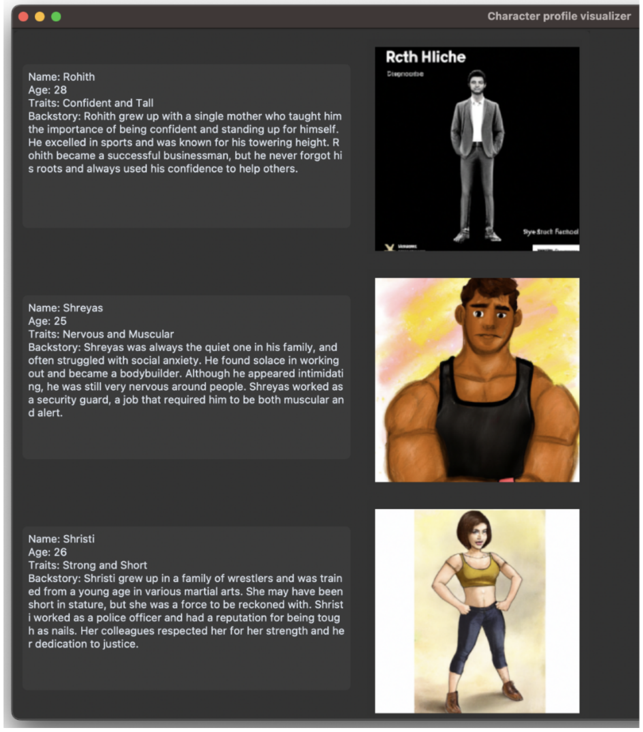
Rohith: Shristi, do you ever think about that day? The explosion, the man we caught?

Shristi: Yeah, I think about it sometimes. It was such a surreal experience.

Rohith: It was. But you know what? I feel like it brought us closer together. We went through something traumatic, but we got through it.

Shristi: I agree. It's funny how adversity can do that. It's like we have this bond now that we wouldn't have had otherwise.

Figure 2: ShortSL User Interface



Character profile visualizer

Name: Rohith
Age: 28
Traits: Confident and Tall
Backstory: Rohith grew up with a single mother who taught him the importance of being confident and standing up for himself. He excelled in sports and was known for his towering height. Rohith became a successful businessman, but he never forgot his roots and always used his confidence to help others.

Name: Shreyas
Age: 25
Traits: Nervous and Muscular
Backstory: Shreyas was always the quiet one in his family, and often struggled with social anxiety. He found solace in working out and became a bodybuilder. Although he appeared intimidating, he was still very nervous around people. Shreyas worked as a security guard, a job that required him to be both muscular and alert.

Name: Shristi
Age: 26
Traits: Strong and Short
Backstory: Shristi grew up in a family of wrestlers and was trained from a young age in various martial arts. She may have been short in stature, but she was a force to be reckoned with. Shristi worked as a police officer and had a reputation for being tough as nails. Her colleagues respected her for her strength and her dedication to justice.

Figure 3: ShortSL User Interface

3.6 Dialogue Generator and Editor

Once the characters and their attributes have been defined by the Profile creator module, the author can choose to use the In-Game Dialogue Generator and Editor to generate in-game social interaction scenes between two or more characters. The author chooses the characters involved, the genre of the conversation, location of the scene and can also provide a description of the expected emotional/ social outcome. The implemented deep learning language model in our system will generate an in-game dialogue scene (multiple utterance pairs) according to the author's inputs. The author can choose to edit and regenerate the dialogues between the characters if they feel that some part of it is not up to their creative liking.

4 SHORTSL - USER INTERFACE

The user interface of ShortSL is built in Python, powered by interface libraries such as Tkinter. The window in Figure 2 showcases the story and dialogue generation modules of our project. The left side of the window takes text inputs from the user under different fields to produce the required story. The central portion of the window displays the user manual to guide the user, just in case they feel lost and do not know the exact functionalities of each widget.

The right side of the window shows two buttons that redirects and triggers the start of the two other modules - Character Image Generation and Dialogue Generation. The Character Image Generation Module, once triggered, will pop up a new window where each character's profile from the story will be displayed. The character's personal details(Name, Age, Traits and Backstory) appears on the left of the profile card while the AI generated animation of the characters appear on the right of the profile card. The Dialogue Generation Module, once triggered, will pop up a new window where the user can input text details on their expectations on how the dialogue should be generated. The social dialogue between two or more characters is generated in the screen below the text inputs.

	ShortSL (GPT-3)	GPT-2 / BERT / RoBERTa
Perplexity	10-20	20-30
BLEU	>=0.8	0.7-0.8
ROUGE	>=0.4	0.3-0.4

Table 1: Performance evaluation of ShortSL for dialogue generation task

Both the story and dialogues that have been generated can be edited using the edit button that will trigger an edit call to the model by taking the existing story into account.

5 Evaluation

The evaluation of performance of ShortSL(GPT-3) was carried out based on its story generation capabilities using two standard datasets, Story Cloze Test and the ROCStories dataset. Story Cloze Test requires a system to choose the correct ending to a four-sentence story and ROCStories requires the same for five-sentence commonsense stories. Coherence, an important parameter in narrative to ensure logical flow was tested with Story Cloze dataset which required generating a plausible ending to a short story. ShortSL achieved an accuracy of 88.4%, while RoBERTa has achieved an accuracy of 86.3%. Diversity, which refers to inclusion of a range of perspectives, experiences, and identities in the stories that are automatically generated was tested with ROCStories dataset which showed ShortSL to achieve a diversity score of 0.72, while RoBERTa achieved a diversity score of 0.64. On the ROCStories dataset, ShortSL backend (GPT3) has achieved a perplexity of 13.67 and F1 score of 0.19 on the task of generating a coherent sixth sentence while BERT has achieved a perplexity of 19.22 and F1 score of 0.11.

Moreover, for evaluating the performance of ShortSL (GPT-3) on the dialogue generation tasks, we compared the performance ShortSL (GPT-3) with GPT-2, BERT, and RoBERTa for three different metrics: Perplexity, BLEU, and ROGUE. Perplexity refers to the inability to deal with or understand something complicated. In terms of NLP, it measures how well a language model is able to predict the next word in a sequence of words. So, the lower the score, the better it is. BLEU(Bilingual Evaluation Understudy) is metric for measuring precision. It signifies how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries. On the other hand, ROGUE(Recall-Oriented Understudy for Gisting Evaluation) is metric for measuring recall. It signifies how much the words (and/or n-grams) in the human reference summaries appeared in the machine-generated summaries. Table 1 summarizes the quantitative metrics for this comparison.

ShortSL (GPT-3) achieved very low perplexity scores on dialogue generation tasks, typically around 10-20. In comparison, other language models such as GPT-2, BERT, and RoBERTa have achieved perplexity scores around 20-30 on similar tasks, indicating that they underperform compared to ShortSL (GPT-3). ShortSL (GPT-3) achieved high BLEU scores on dialogue generation tasks, typically around 0.8 or higher. Other language models such as GPT-2, BERT, and RoBERTa have achieved BLEU scores around 0.7-0.8 on similar tasks, indicating that they underperform slightly compared to ShortSL (GPT-3). ShortSL (GPT-3) achieved high ROUGE scores on dialogue generation tasks, typically around 0.4 or higher. Other language models such as GPT-2, BERT, and RoBERTa have achieved ROUGE scores around 0.3-0.4 on similar tasks, indicating that they underperform slightly compared to ShortSL (GPT-3).

Overall, ShortSL (GPT-3) has demonstrated strong performance on story generation and dialogue generation tasks, outperforming other popular language models such as GPT-2, BERT, and RoBERTa. However, it is worth noting that the relative performance of different language models can vary depending on the specific task being evaluated, and it is important to evaluate each language model on its own merits for each specific task.

6 Novelty and Value

Most story generation systems work upon a set of predefined inputs from the author that are mandatory for their functioning. Here the author can choose to input whatever they have come up with instead of restricting themselves to a given set of mandatory inputs. This enhances creative freedom and accommodates the different ways of thinking employed by different authors(some may prefer to start the story by writing the climax while some may define the genre or characters first). Hence, this framework is very adaptive to the type of author using it in terms of creativity and enables freedom in their thought processes. The story can also be dynamically regenerated upon as many edits as the author requires.

Character profiles and backstories are created dynamically based upon the main storyline and initial inputs of the author, which is not usually integrated in past systems and provides authors with further understanding over even the randomly generated characters(if any) so that the authors can choose to modify their attributes according to their original creative ideas.

Apart from the story generation process, This system aims to provide the author with control of the visual aspects of the game as well in addition to its story generation capabilities.The character visualization module helps show the predicted physical appearance of characters based upon the author's writing. This can be used by the author to give the animators the idea of what the original character is supposed to look like, based upon the author, providing them with a certain level of authority over the non writing part of the game creation process as well.

The dynamic in-game dialogue generation has also been proposed based upon social interactions between two or more characters where the author can choose location and describe the outcome of the dialogue. It aims to act as a one stop solution for assisting authors in the initial creative processes before developing an interactive game.

7 Future work

In the future, the Short Story Long (ShortSL) project can be extended to incorporate more advanced natural language processing techniques that can help the system better understand and analyze the author's input to generate more realistic and compelling storylines. Additionally, the system can be trained on a larger dataset of existing stories and games to improve its ability to generate diverse and innovative ideas.

The system can also be integrated with existing game engines to allow for real-time testing and integration of generated storylines and characters into the game. Furthermore, user feedback can be incorporated into the system to improve its performance and adaptability to different authors' needs and thinking processes.

Moreover, we know that the current system provides the functionality of dynamically editing and regenerating a new story/dialogue conversation if it doesn't match an author's creative liking. One potential future work in this direction is to improve ShortSL to provide this functionality even for the generated character images too, and then the authors would have the ability to keep regenerating visualizations of the characters until they are completely satisfied and feel that it suits well in the entire narrative that they have envisioned.

Furthermore, ShortSL currently relies on inputs from the author, but future work could explore ways to incorporate more diverse input data, such as images or audio recordings. This could potentially improve the accuracy and diversity of the generated storylines and characters.

Lastly, ShortSL could be further developed to allow for collaborative storytelling, where multiple authors can work together on the same project. This could involve implementing features such as version control or a shared workspace for authors to brainstorm and share ideas and consider all authors' creative thinking abilities while generating the storyline/character profiles/dialogue conversations.

8 Conclusion

Our proposed project aims to provide a complete integrated framework that can assist any author regardless of their thinking process in most idea generation/brainstorming/creation processes involved for creating a game's storyline, characters, appearances, in-game dialogues etc. This helps accommodate authors of different thinking characteristics on a single automated platform to help in story generation, character profile creation. Authors and Idea creators for interactive and narrative games will find this tool extremely helpful. Some challenges and future works for this project include targeted improvements take positive steps towards achieveing maximal narrative coherence, emotional resonance, story originality, continual adaptation and flexibility.

9 Code and Documentation

All the code bases related to this project have been uploaded in the GitHub repository `https://github.com/rohis06/ECS289G-project` for a collaborative environment. For ease of communication across the group, we have used Slack and Discord. All the documentation related to this project was done on Google Docs due to its shareable nature. A README.md file has also been provided in the GitHub repo to give an overview of the project.

References

[1] A. Alabdulkarim, S. Li, and X. Peng, "Automatic story generation: Challenges and attempts," 2021.

[2] B. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," pp. 239–253, 2017.

[3] J. Skorupski, L. Jayapalan, S. Marquez, and M. Mateas, "Wide ruled: A friendly interface to author-goal based story generation," Berlin, Heidelberg, pp. 26–37, 2007.

[4] I. Grabska-Gradzińska, L. Nowak, W. Palacz, and E. Grabska, "Application of graphs for story generation in video games," New York, NY, USA, 2021. [Online]. Available: https://doi.org/10.1145/3437378.3442693

[5] D. Liu, J. Li, M.-H. Yu, Z. Huang, G. Liu, D. Zhao, and R. Yan, "A character-centric neural model for automated story generation," pp. 1725–1732, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5536

[6] F. Nunnari and A. Heloir, "Generation of virtual characters from personality traits," Cham, pp. 301–314, 2017.

[7] A. I. Alhussain and A. M. Azmi, "Automatic story generation: A survey of approaches," New York, NY, USA, may 2021. [Online]. Available: https://doi.org/10.1145/3453156

[8] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, "Towards the automatic anime characters creation with generative adversarial networks," 2017.

[9] N. Peng, M. Ghazvininejad, J. May, and K. Knight, "Towards controllable story generation," New Orleans, Louisiana, pp. 43–49, Jun. 2018. [Online]. Available: https://aclanthology.org/W18-1505

[10] F. Palmas, J. Raith, and G. Klinker, "A novel approach to interactive dialogue generation based on natural language creation with context-free grammars and sentiment analysis," pp. 79–83, 2020.

[11] C. Kerr and D. Szafron, "Supporting dialogue generation for story-based games," pp. 154–160, Oct. 2009. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/12371

[12] J. Cho, M. Jeong, J. Bak, and Y.-G. Cheong, "Genre-controllable story generation via supervised contrastive learning," New York, NY, USA, p. 2839–2849, 2022. [Online]. Available: https://doi.org/10.1145/3485447.3512004