ECE1786 Final Report

GOTalk

Team Member: Tuoyue Huang; Yezheng Shao

Outline

Introduction **2**

Illustration **2**

Background & Related Work **3**

Data Processing **3**
- Data Source and Target 3
- Cleaning Steps 3

Architecture and Software **4**
- GPT2 4
  - Context Generation 5
  - Dialogue Generation 5
- GPT3 6
  - Generation 7
  - Solution 7
- Software 7

Quantitative Results **8**
- Evaluation standard of model generation 8

Qualitative Results **9**
- GPT2 9
- GPT3 9

Discussion and Learnings **12**

Individual Contributions **13**
- Tuoyue 13
- Yezheng 13

Reference **14**

Appendix **14**

Permission **15**

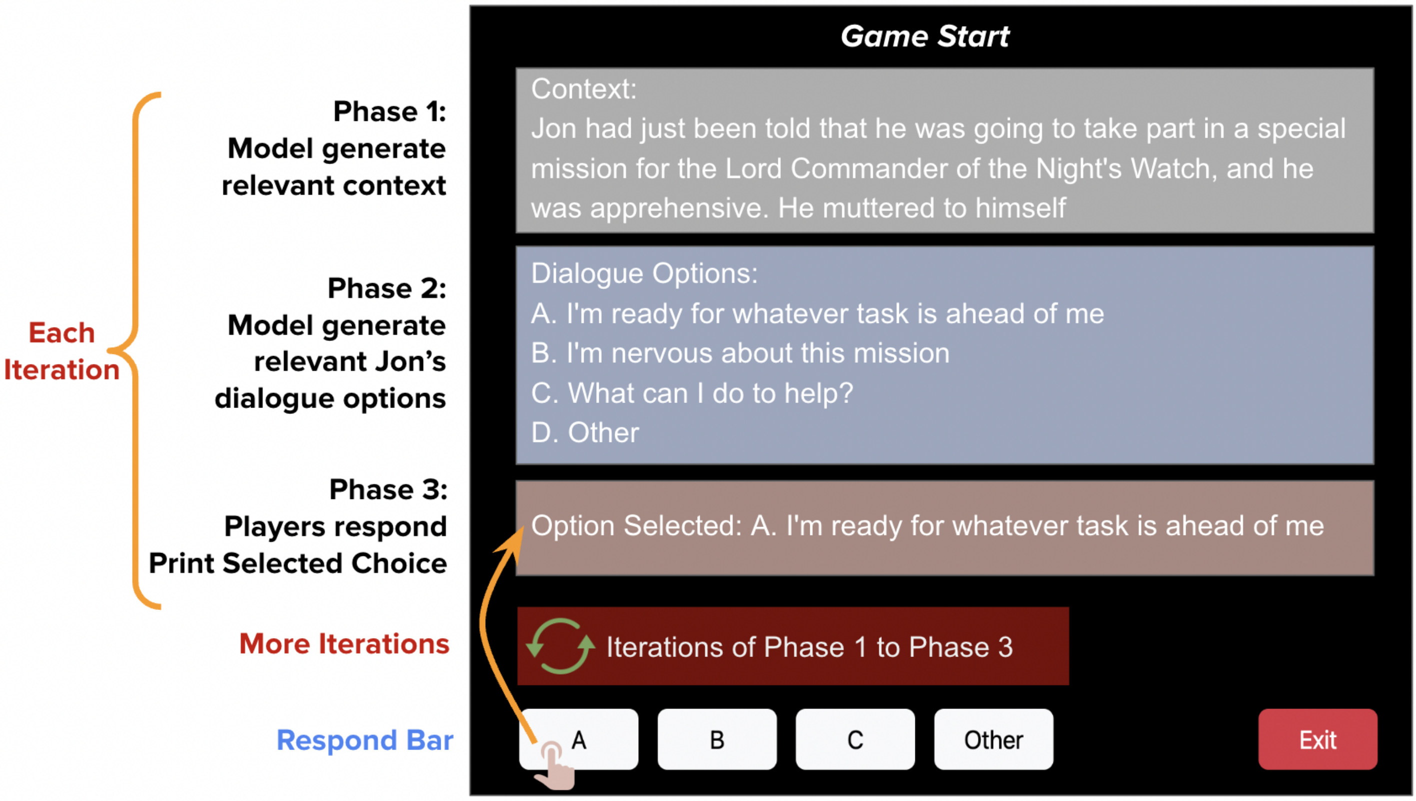
Word count: 2000

Introduction

The project aims to design a role-playing text game based on Generative Pre-trained Transformers (GPT). The game will be set in the "Game of Thrones" (GOT) world. Players will take on the role of Jon Snow. They will read the context and choose one of the three dialogue options for Jon as the response to influence the following context.

This project is mainly for fun, where players could have the experience of being involved in the GOT story. Additionally, if the model is fine-tuned with other specific context data, it could serve as a reference tool for the plot design of stories or movies.

Illustration



Game Start

Context:
Jon had just been told that he was going to take part in a special mission for the Lord Commander of the Night's Watch, and he was apprehensive. He muttered to himself

Dialogue Options:
A. I'm ready for whatever task is ahead of me
B. I'm nervous about this mission
C. What can I do to help?
D. Other

Option Selected: A. I'm ready for whatever task is ahead of me

Iterations of Phase 1 to Phase 3

Fig. 1

Background & Related Work

The paper[1] is focused on the automatic generation of interactive stories. Rather than simply generating texts with the appended user-chosen input, the paper tries to improve story consistency by building short-term and long-term memory. This is done by analyzing the input paragraphs with the Named Entity Recognition and WordNet Synsets Categorization, so a reasonable set of actions would be formed from the memories.

Our goal is to create an interactive dialogue game with the background of Game Of Thrones. AI Dungeon[2] is a deployed project that is very similar to ours. It is a text-generation adventurous game. The project used the GPT-2 model that has 1.5B parameters and fine-tuned it with a collection of text adventures. Their project allows users to input any action and the model will generate a story based on that action.

Data Processing

Data Source and Target

The data are obtained from Kaggle Datasets, which contain the text scripts of all the books in the "A Song of Ice and Fire" series. This provides both the narratives and dialogues of the GOT story.

In order to play as Jon Snow and speak for him, we need to locate the dialogues of Jon and mark them for our model to learn. However, dialogues in books are often written in a more vivid manner than simply stating who said what. The speaker's name can be written before, within, or after the quotation marks. Therefore, we need to find and transform Jon's dialogues into the who-said-what format. This is vital for improving the generation of complete and meaningful sentences in the game.

Cleaning Steps

1. Filter sections of Jon from all the books and merge them together, which has around 4900 lines of text
2. Find and transform Jon’s dialogues & add special dialogue tokens:
   We have written an alterline function to process the text line by line to deal with the following three cases:

   ➢ **Case 1:**
   - One pair of quotation marks
   - ‘Jon’ is at the start of the line.
     - Example: Jon’s throat was dry. “You know?”
     - Transformed: Jon’s throat was dry. [BOS] You know? [EOS]

- **Case 2:**
    - One pair of quotation marks
    - ‘Jon’ is right after the first ending quotation and followed by a past verb ending with ‘ed’ or is in the list ["said", "felt", "told"]
        - Example: “I have to,” Jon said fervently.
        - Transformed: [BOS]I have to, [EOS] Jon said fervently.
- **Case 3:**
    - Two pairs of quotation marks
    - Same condition as Case 2
        - Example: “No,” Jon said at once. “That was…”
        - Transformed: Jon said at once. [BOS] No, That was… [EOS]

3. Delete the rest of the quotation marks
4. Only keep lines with [BOS] and [EOS] & remove symbols around the tokens
5. Split data into Train and Test files with a ratio of 0.2
6. Tokenized input data for the model with a block size of 50

Finally, we get 378 training samples and 98 testing samples for each epoch

Decoded examples:

- Without special tokens:
  ‘There were times (… omitted…) Jon Snow was glad he was a bastard’
- With special tokens:
  ‘are the (…omitted…). Jon admitted stiffly [BOS] Lord Eddard’

Architecture and Software

GPT2

We found a pre-trained GOT model from the hugging face, which is the GPT2-large model with 36-layer, 1280-hidden, 20-heads and 774M parameters. It could produce GOT-related content, but it cannot directly fulfill the goal of our project. So we fine-tuned the pre-trained model with our dataset.

The fine-tuned arguments are set so that the model could generate valid contexts and dialogues. Some key arguments are shown as follows:

- num_train_epochs = 3
- per_device_train_batch_size = 2
- per_device_eval_batch_size = 2
- learning_rate = 9e-6
- eval_steps = 50

Despite fine-tuning, another vital part would be how the generations of context and dialogue are done for our game. We used two critical parameters in the model generate function, which are stopping_criteria and bad_words_ids, to ensure the generation is appropriate and relevant. The stopping criteria determine when the model should stop generating text, while the bad word specifies a list of words that the model should avoid using during the generation.

A pre-set tokenized introduction would be the first model input, like ‘Winter is coming ’, to initialize the game iteration. For the rest inputs, we just append the generated contexts or selected dialogues to the previous input and use the last 50 tokenized words as the final input.

Context Generation

- stopping_criteria: [BOS]
- bad_words_ids: [EOS]
- min_length = 60
- max_new_tokens = 150
- temperature = 0.95
- top_p = 1
- repetition_penalty = 1.1

Raw output example:

“ <u>Model Input Ends With [EOS]</u> winter is coming for all of us. Jon said [BOS] ”

Presented in Phase 1:

“ winter is coming for all of us. Jon said ”

With the setting, the generation will automatically end when the model predicts the [BOS] and the [EOS] won’t appear in the generation. This gives us the desired context at a proper length for each context phase.

Dialogue Generation

- stopping_criteria : [EOS]
- bad_words_ids : [BOS] , " , Jon , Snow , he
- min_length = 3
- max_new_tokens = 50
- temperature = 1
- top_p = 1
- repetition_penalty = 1.1

Raw output example:

“ <u>Model Input Ends With [BOS]</u> What is your name, boy? [EOS] ”

One option presented in Phase 2:

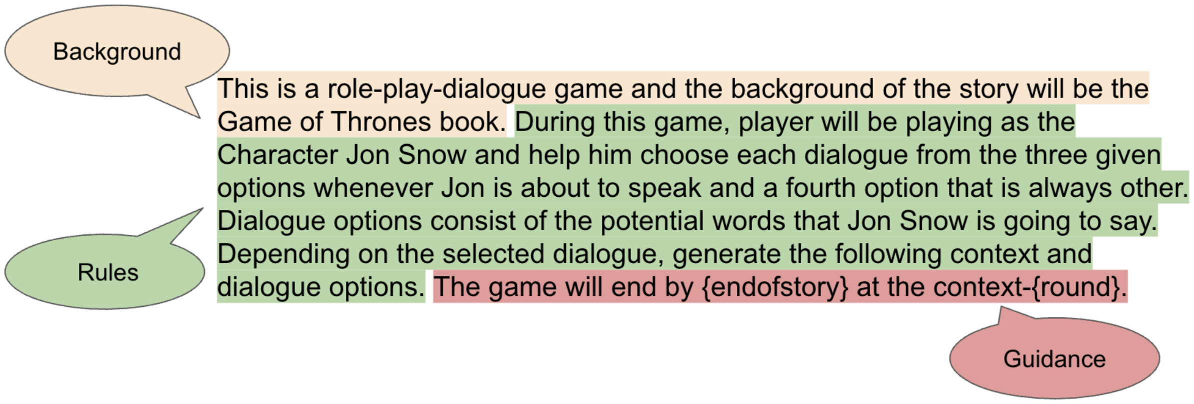
“ What is your name, boy? ”

Similarly, the generation stops when the prediction is [EOS] and the generation repeats three times to get us three dialogue options with the same input. Words in bad_words_ids help to predict better dialogues, for instance, Jon usually will not claim himself as ‘Jon’ or ‘Snow’. While the word ‘he’ usually leads the generation to be more context-like, so we exclude it.

GPT3

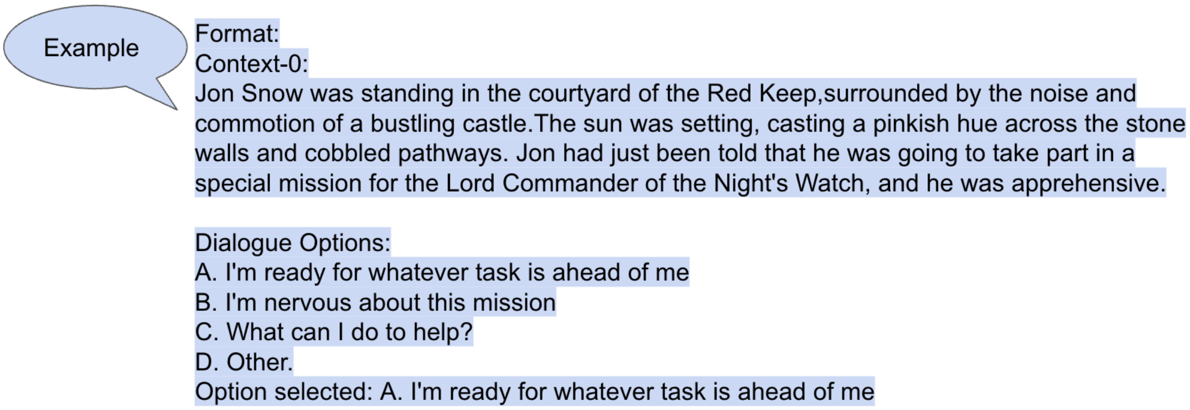
For our advanced model, we use the GPT3 text-completion model. We first give input to the model as the prompt, and then the model will generate a text completion that matches the given prompt.

The prompt includes the background, rules, and guidance for the game, as well as an example of one-shot learning used as the starting context for the game. As shown in the figure below:


Background
Rules
Guidance

This is a role-play-dialogue game and the background of the story will be the Game of Thrones book. During this game, player will be playing as the Character Jon Snow and help him choose each dialogue from the three given options whenever Jon is about to speak and a fourth option that is always other. Dialogue options consist of the potential words that Jon Snow is going to say. Depending on the selected dialogue, generate the following context and dialogue options. The game will end by {endofstory} at the context-{round}.

Fig. 2



Format:
Context-0:
Jon Snow was standing in the courtyard of the Red Keep,surrounded by the noise and commotion of a bustling castle.The sun was setting, casting a pinkish hue across the stone walls and cobbled pathways. Jon had just been told that he was going to take part in a special mission for the Lord Commander of the Night's Watch, and he was apprehensive.

Dialogue Options:
A. I'm ready for whatever task is ahead of me
B. I'm nervous about this mission
C. What can I do to help?
D. Other.
Option selected: A. I'm ready for whatever task is ahead of me

Fig. 3

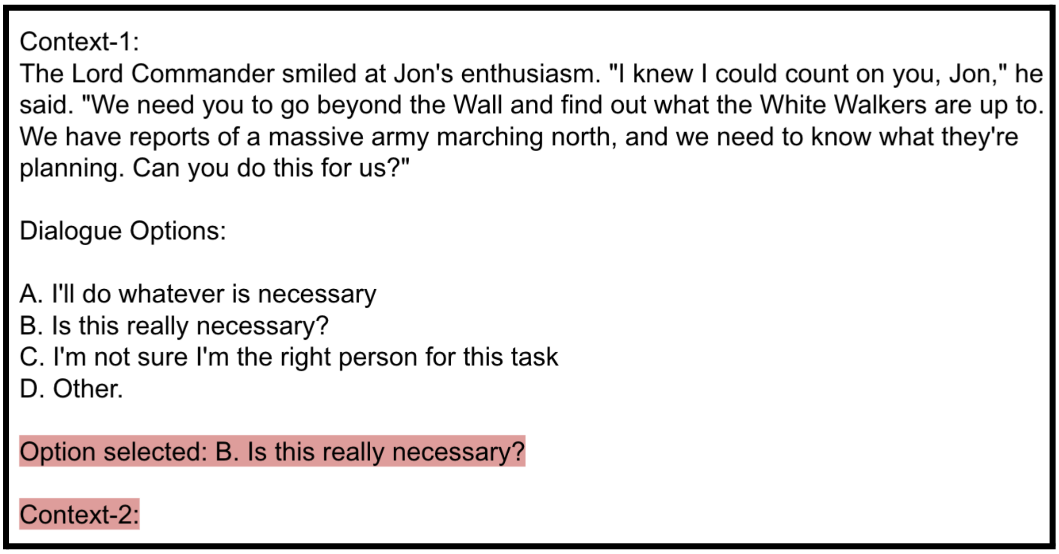
Generation

- temperature: 1
- max_token: 512
- top_p: 0.9
- frequency_penalty: 0.3
- presence_penalty: 0.2
- Stop: “Option selected”
- best_of: 3

In the generation phase, we set the stop sequence to “Option selected:” to stop the generation before the player selection. After the player selects the dialogue, that dialogue will be appended after the “Option selected”. Then we will feed all the generated so far plus the original prompt as the prompt of the next generation. Our expectation of the model is that it will follow the format in the example, however, during testing we find that the model sometimes will skip the context or dialogue options and end the story immediately.

Solution

To solve this instability in the generation, we input the context header manually each round to give the model a hint that the following generation should match the example format.



Context-1:
The Lord Commander smiled at Jon's enthusiasm. "I knew I could count on you, Jon," he said. "We need you to go beyond the Wall and find out what the White Walkers are up to. We have reports of a massive army marching north, and we need to know what they're planning. Can you do this for us?"

Dialogue Options:

A. I'll do whatever is necessary
B. Is this really necessary?
C. I'm not sure I'm the right person for this task
D. Other.

Option selected: B. Is this really necessary?

Context-2:

Fig. 4

The highlight section in the figure above is the part that is manually added. We have a counter run in the background that counts the number of contexts generated so far. Therefore, we can provide the correct context header for the next generation based on the counter.

Software

We implemented our software with React and Flask. We also deployed it on the Amazon EC2 instance.

Quantitative Results

Evaluation standard of model generation

Incorrect Structure:
- context too short and not meaningful
- no [BOS] or [EOS] was predicted when generation reaches the maximum
- inappropriate position of [BOS]
- no context or dialogue has been generated

Incorrect Dialogue:
- more context-like
- clearly not speak by Jon
- repetitive dialogue options

Incoherent Generation:
- some grammatical errors and not conform to common logic
- contextual disconnect

	# Incorrect Structure	# Incorrect Dialogue	# Incoherent Generation	# Total Incorrect	# Total Generation
GPT2	32	3	9	44	100
GPT3	2	1	0	3	100

Table 1.

Since our task is text generation, it is hard for us to evaluate with common metrics. Therefore, we set our standard based on the game we want to realize. Both models generate 100 times, and we manually judge each generation accordingly, giving us the table above. Thus, GPT2 has an error rate of 44%, while GPT3 only has 3%. Most incorrect generations of GPT2 are caused by the first two rules from Structure and incoherent generation. Its dialogue generation actually goes well.

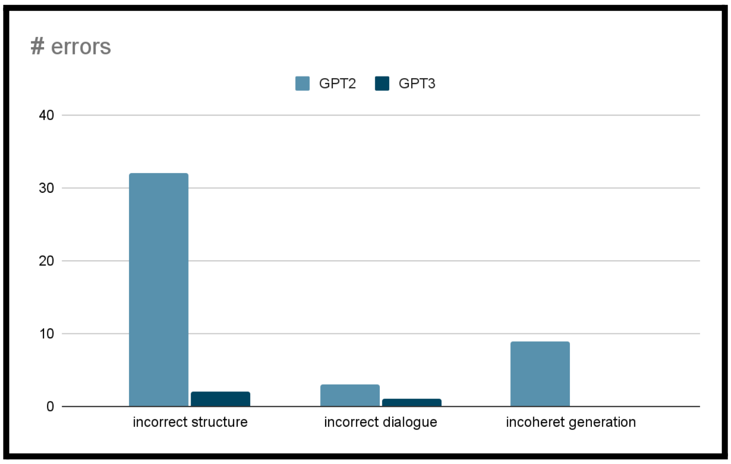
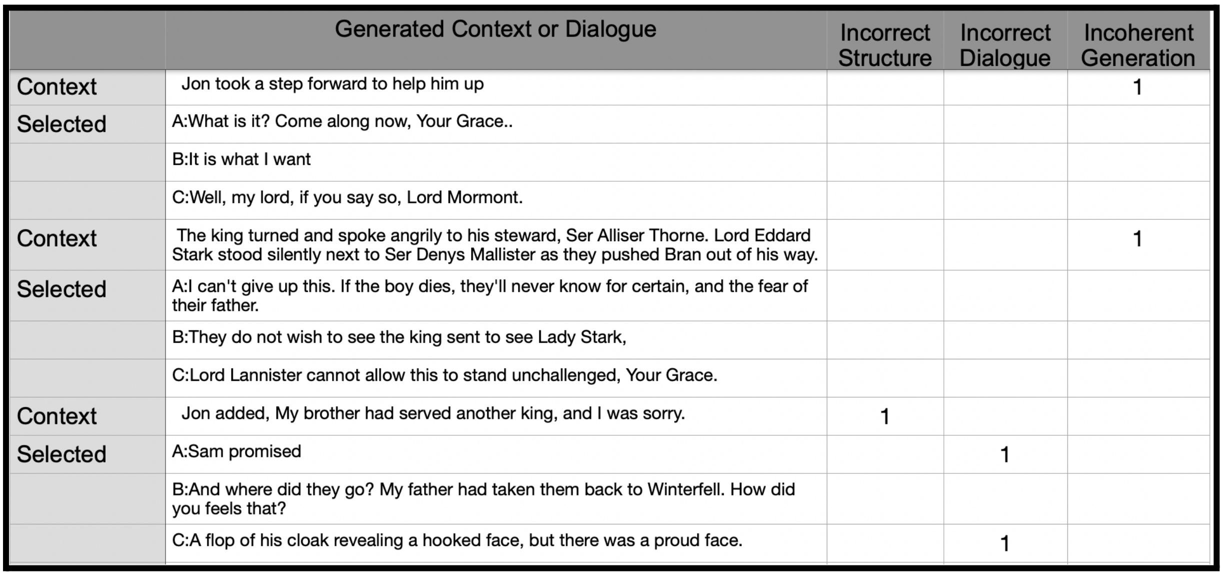


Fig. 5

Qualitative Results

GPT2

Overall performance is not good for GPT2, we select three iterations from the above 100 generations. We observe that the dialogue generation usually performs well, except for the last A and C, which are more context-like. All three contexts are problematic, they are usually incoherent from the selected choice. Sometimes, like in the last context, 'My brother … sorry' sounds more like to be dialogue said by Jon instead of showing up in the context.

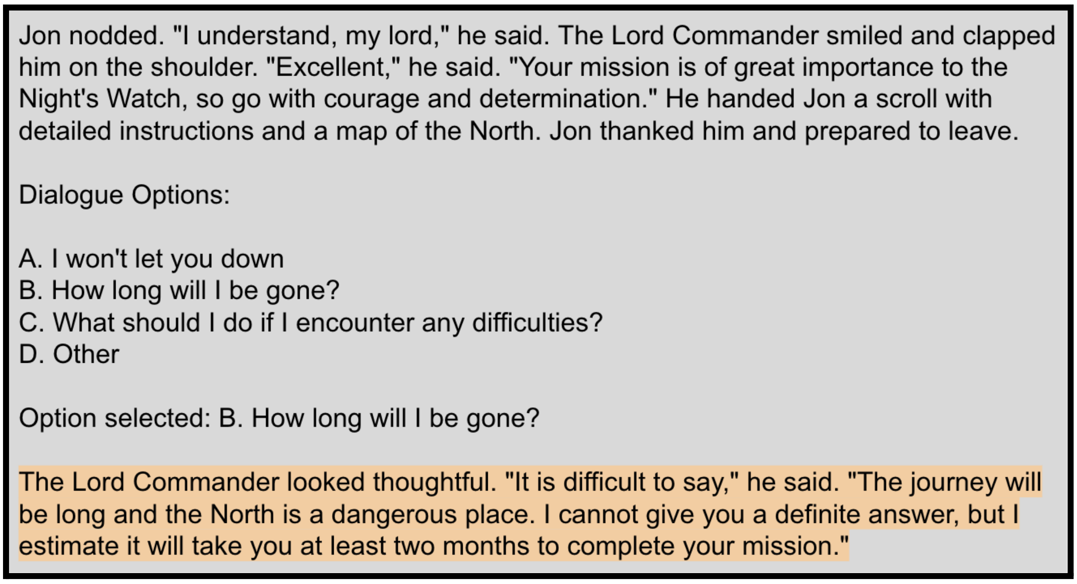


	Generated Context or Dialogue	Incorrect Structure	Incorrect Dialogue	Incoherent Generation
Context	Jon took a step forward to help him up			1
Selected	A:What is it? Come along now, Your Grace..			
	B:It is what I want			
	C:Well, my lord, if you say so, Lord Mormont.			
Context	The king turned and spoke angrily to his steward, Ser Alliser Thorne. Lord Eddard Stark stood silently next to Ser Denys Mallister as they pushed Bran out of his way.			1
Selected	A:I can't give up this. If the boy dies, they'll never know for certain, and the fear of their father.			
	B:They do not wish to see the king sent to see Lady Stark,			
	C:Lord Lannister cannot allow this to stand unchallenged, Your Grace.			
Context	Jon added, My brother had served another king, and I was sorry.	1		
Selected	A:Sam promised		1	
	B:And where did they go? My father had taken them back to Winterfell. How did you feels that?			
	C:A flop of his cloak revealing a hooked face, but there was a proud face.		1	

Fig. 6

GPT3

Example of Incorrect Structure



Jon nodded. "I understand, my lord," he said. The Lord Commander smiled and clapped him on the shoulder. "Excellent," he said. "Your mission is of great importance to the Night's Watch, so go with courage and determination." He handed Jon a scroll with detailed instructions and a map of the North. Jon thanked him and prepared to leave.

Dialogue Options:

A. I won't let you down
B. How long will I be gone?
C. What should I do if I encounter any difficulties?
D. Other

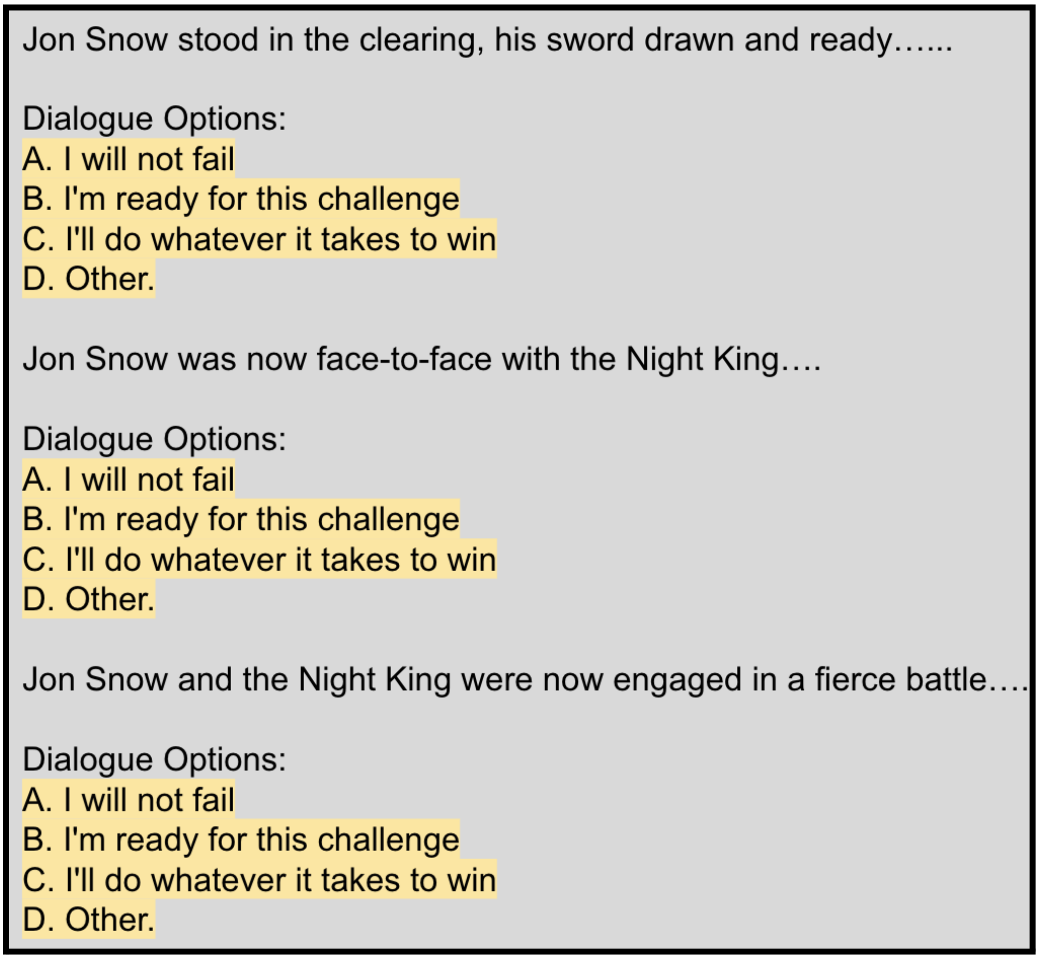
Option selected: B. How long will I be gone?

The Lord Commander looked thoughtful. "It is difficult to say," he said. "The journey will be long and the North is a dangerous place. I cannot give you a definite answer, but I estimate it will take you at least two months to complete your mission."

Fig. 7

As you can see from the figure above, the generation stopped after the context. This is a bad generation since there are no available options for players to choose from.

Example of Incorrect Dialogue



Jon Snow stood in the clearing, his sword drawn and ready……

Dialogue Options:
A. I will not fail
B. I'm ready for this challenge
C. I'll do whatever it takes to win
D. Other.

Jon Snow was now face-to-face with the Night King….

Dialogue Options:
A. I will not fail
B. I'm ready for this challenge
C. I'll do whatever it takes to win
D. Other.

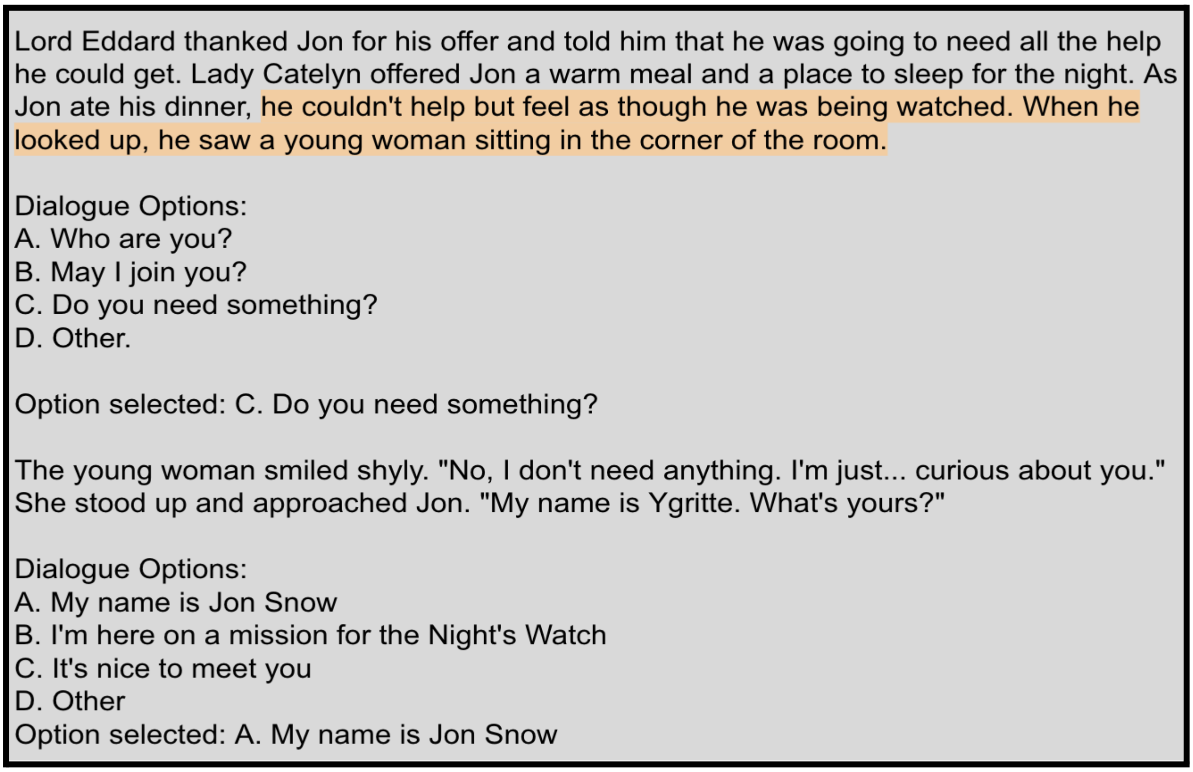
Jon Snow and the Night King were now engaged in a fierce battle….

Dialogue Options:
A. I will not fail
B. I'm ready for this challenge
C. I'll do whatever it takes to win
D. Other.

Fig. 8

To save some space, we have truncated the context and the selected option. But from the highlighted dialogue options, we can see that these three consecutive contexts give the same options. These repetitive dialogue options show that the story doesn't make enough progress and it will easily bore the players.

Example of Good Generation



Lord Eddard thanked Jon for his offer and told him that he was going to need all the help he could get. Lady Catelyn offered Jon a warm meal and a place to sleep for the night. As Jon ate his dinner, he couldn't help but feel as though he was being watched. When he looked up, he saw a young woman sitting in the corner of the room.

Dialogue Options:
A. Who are you?
B. May I join you?
C. Do you need something?
D. Other.

Option selected: C. Do you need something?

The young woman smiled shyly. "No, I don't need anything. I'm just... curious about you." She stood up and approached Jon. "My name is Ygritte. What's yours?"

Dialogue Options:
A. My name is Jon Snow
B. I'm here on a mission for the Night's Watch
C. It's nice to meet you
D. Other
Option selected: A. My name is Jon Snow

Fig. 9

The above example is generated when the player sets the goad to “Jon Snow finds his love of life”. As we can see from the highlighted section, the model generates a plot that Jon Snow met this woman. This creates a smooth change that leads the story toward the desired ending.

Discussion and Learnings

Overall, the GPT3 text-completion model performs much better than our fine-tuned GPT2 model. GPT2 consistently fails to produce a relevant context and the transition of characters is not smooth, likely due to its smaller model size and fewer training examples.

When it comes to GPT3, it surprises us with a nearly perfect performance. The model is able to understand our prompts and learn from examples, resulting in a 97% success rate for generating valid output. One thing that surprised us was its relevance to the book Game Of Thrones. For instance, when we select the ending to be "Jon Snow finds his love of life", the generated story leads Jon Snow to meet Ygritte.

We learned a lot from this project. The most important takeaway is the importance of cleaning raw data and formatting it properly for the model. This requires adding special tokens to the data so the model can recognize it. We also learned about how to use advanced models like GPT3 to create engaging games. If given the opportunity to start a similar project, we would start by using the most advanced but feasible model available. On the other way around, we could take the advantage of few-shot-learning with GPT3 to create the demanded data since we have observed its amazing performance.

Individual Contributions

Tuoyue

- Merged all Jon’s sections
- Coded dialogue Case 1
- Cleaned the punctuations
- Finalized the preprocessing pipeline
- Split and tokenized dataset for GPT2
- Finetuned GPT2 and found best generating parameters
- Plot training curves
- Finalized the GPT2 game generating engine
- Evaluated 100 generations from GPT2 game

Yezheng

- Downloaded data from Kaggle
- Extracted Jon’s sections from books
- Coded dialogue case 2 and 3
- Engineered GPT3 prompt and found best parameters
- Implemented the interactive software using Flask and React
- Deployed the software on EC2
- Evaluated 100 generations from GPT3 game

Reference

[1]
J. Freiknecht and W. Effelsberg, “Procedural Generation of Interactive Stories using Language Models,” *International Conference on the Foundations of Digital Games*, Sep. 2020, doi: 10.1145/3402942.3409599.
[2]
N. Walton, “AI Dungeon 2: Creating Infinitely Generated Text Adventures with Deep Learning Language Models,” *Perception, Control, Cognition*, Nov. 21, 2019. https://pcc.cs.byu.edu/2019/11/21/ai-dungeon-2-creating-infinitely-generated-text-adventures-with-deep-learning-language-models/ (accessed Dec. 13, 2022).

Appendix

1. Data source: https://www.kaggle.com/datasets/khulasasndh/game-of-thrones-books?select=004ssb.txt

2. Pretrained GPT2 model — HScomcom/gpt2-game-of-thrones: https://huggingface.co/HScomcom/gpt2-game-of-thrones?text=My+name+is+Lewis+and+I+like+to

3. Final fine-tuned GPT2 model —- huangtuoyue/GPT2-large-GOTfinetuned_v5: https://huggingface.co/huangtuoyue/GPT2-large-GOTfinetuned_v5

4. Complete GPT2 100 game generations: https://github.com/ece1786-2022/GOTalk/blob/main/Evaluation/GPT2_Result_Evaluation.numbers

5. Complete GPT3 100 game generations: https://github.com/ece1786-2022/GOTalk/blob/main/Evaluation/GPT3_evaluation_text.txt

Permission

Tuoyue

- permission to post video: wait till see video
- permission to post final report: yes
- permission to post source code: yes

Yezheng

- permission to post video: wait till see video
- permission to post final report: yes
- permission to post source code: yes