JEDI: Justifiable End-dialogue Driven Interaction for NPC Entities in Role-Playing Games

Stanford CS224N Custom Project

Willy Chan
Department of Computer Science
Stanford University
willyc@stanford.edu

Omar Abul-Hassan
Department of Computer Science
Stanford University
omarah@stanford.edu

Sokserey Sun
Department of Computer Science
Stanford University
sokserey@stanford.edu

Abstract

In the realm of story-driven role-playing games (RPGs), generating dynamic, context-sensitive dialogue is pivotal for enhancing player immersion and engagement. Traditional methods rely on pre-written scripts, which are labor-intensive and limited in adaptability. JEDI investigates the potential of large language models to revolutionize RPG dialogue generation, using the game "Star Wars: Knights of the Old Republic" (KOTOR) as a case study. We preprocess the KOTOR dataset to graph and linearize dialogue sequences alongside game state information, then fine-tune state-of-the-art LLMs (BART, GPT-2, GPT-3.5 Turbo) to generate contextually relevant and engaging dialogue. Our evaluation, employing metrics like BLEURT, BERTScore, and DialogueRPT, demonstrates significant improvements in generating coherent and contextually appropriate dialogue when fine-tuning. The final fine-tuned BART model's BLEURT score improved from -1.3090 to -0.9215, while GPT-3.5 Turbo achieved a BERTScore of 0.8940. Additionally, GPT-3.5 Turbo's DialogRPT scores for human-vs-rand and human-vs-machine were 0.5118 and 0.999, respectively. The introduction of cross-attention in GPT-2 further enhanced its performance, with BLEURT improving from -0.8000 to -0.7014, and achieving a BERTScore of 0.8535. This work contributes a framework for integrating LLMs in branching, narrative-based RPGs, paving the way for more interactive and immersive game narratives.

1 Key Information to include

Our mentor was Rashon Poole, we had **no** External Collaborators, and we are **not** sharing this project. Willy worked on fine-tuning BART and evaluated it's results, and created the dataset. Sokserey worked on fine-tuning GPT 3.5 and evaluated it. Omar added cross attention to GPT2 and examined the results. All members played a significant role in writing the report.

2 Introduction

Video games are a rapidly expanding sector within the entertainment industry, with story-driven role-playing games (RPGs) standing out as some of the most popular and financially successful genres globally, estimated to generate billions of dollars in revenue annually (Baltezarevic et al. (2018)). RPGs are characterized by their complex narrative structures and the extensive autonomy they provide players to shape the game's outcome through their choices and playstyles.

Most contemporary RPGs feature text and dialogue crafted by human writers—a process that requires significant time and financial investment. This allows the dialogue to reflect various game states and player interactions, making for a more immersive experience. However, the emergence of large language models offers a new frontier for narrative interactivity; these models can potentially generate dynamic, context-sensitive dialogue in response to freeform inputs: providing a customized experience that adapts to player-generated input, choices and state. Such a system could revolutionize the way narratives are integrated into games, making them more interactive and immersive.

However, implementing effective dialogue generation within the complex, state-driven framework of RPGs poses significant technical hurdles. These challenges stem from the need to maintain coherent and contextually appropriate interactions over extended dialogue sequences, which can vary dramatically based on player choices and game events (Akoury et al. (2023)).

In response to these challenges, our project develops an end-to-end dialogue generation system tailored for the interactive narratives of RPGs. By preprocessing the Star Wars: Knights of the Old Republic (KOTOR) dataset, we innovate on existing methodologies by graphing and linearizing dialogue sequences alongside game state information, and then fine-tune pretrained language models to fill in masked portions of the conversation. This allows our system to maintain narrative coherence across player interactions while adapting to the evolving game environment. Our approach leverages the strengths of multiple state-of-the-art language models, namely BART, GPT-2, and GPT-3.5 Turbo, which are fine-tuned to generate contextually relevant and engaging dialogue.

3 Related Work

Previous work by Akoury et al. (2023) explores methods of generating relevant datasets, using the video game *Disco Elysium* as a case-study. The study lays the groundwork for deriving datasets relevant to training open-ended dialogue models. In particular, this specific dataset is structured as a graph, where nodes represent possible utterances and edges denote transitions based on the game state, which is encoded in Lua scripts. The authors utilize clustering techniques to group similar dialogue nodes, enhancing the coherence of generated responses by considering the game's state. They then linearize these clusters into Lua scripts, masking one utterance at a time to allow LLMs like GPT-3 Curie and Codex to generate plausible alternatives. We adapt a similar approach for the KOTOR dataset, by similarly clustering and linearizing conversation sequences.

The availability of text corpora for RPGs is extensive and diverse, providing a valuable resource for research and development in the field of dynamic dialogue generation. Studies like those conducted by van Stegeren and Theune (2020) highlight multiple RPG corpora, each containing hundreds of thousands of tokens. These datasets are derived from popular RPGs and include extensive dialogue sequences, narrative structures, and player interactions, making them ideal for training and fine-tuning language models.

Further work by Värtinen et al. (2024) explores using models like GPT-2 and GPT-3 for generating RPG item descriptions, highlighting the effectiveness of fine-tuning transformer models to produce text acceptable to human readers. Although the task is different, the significant improvements observed from GPT-2 to GPT-3 underscore the potential of advanced models for dynamic content creation. Other promising results, like those found by Zhang et al. (2020), show that LLMs are capable of sustaining long-form conversation-like exchanges: generating content that is relevant and truthful. Inspired by results from Yan et al. (2021) indicating that tuning attention can significantly increase training and inference speed, we were also motivated to adjust the attention mechanisms in our GPT-2 model to improve performance.

4 Approach

We fine-tuned 3 specific models for our work. GPT-2 uses a unidirectional transformer architecture with a multi-head self-attention mechanism and a position-wise feed-forward network, incorporating layer normalization and residual connections. It is trained on extensive internet data, enabling it to generate contextually relevant text (Radford et al. (2019)). In contrast, GPT-3.5, an extension of GPT-3, has significantly more parameters, demonstrating enhanced contextual understanding, few-shot learning capabilities, and superior performance across various NLP tasks (Brown et al. (2020)). BART (Bidirectional and Auto-Regressive Transformers), introduced by Facebook AI, employs an encoder-decoder architecture with a denoising objective, making it versatile for tasks like text generation, translation, and summarization. BART's encoder processes input bidirectionally, while its autoregressive decoder generates output token by token (Lewis et al. (2020)). The code for masking the dataset and fine-tuning BART, and GPT-2 was written by ourselves. The code for calculating the scores for these models and GPT-3.5 was also written by ourselves. We achieved our baseline results with base GPT-2 and BART.

Also, during training, label smoothing was implemented to prevent the model from becoming overly confident about its predictions, particularly for GPT-3.5. We did not implement label smoothing, rather it came with training the model on the GPT API. This technique modifies the one-hot encoded ground truth labels to a softer distribution, defined as:

$$y_{smooth} = y_{true} \cdot (1 - \alpha) + \frac{\alpha}{K}$$

where y_{true} represents the original one-hot encoded label, α is the smoothing parameter, and K denotes the number of classes. This approach ensures that the loss function does not become too confident about the correct class, thus improving generalization.

In addition to the standard GPT-2, we experimented with a modified **original** version incorporating a cross-attention mechanism (GPT2CA) inspired from Gheini et al. (2021). The motivation behind this architectural enhancement was to leverage the available game state variables to improve the model's ability to generate contextually relevant and coherent responses. By paying attention to specific parts of the input sentence that are most relevant to the game state, we hypothesized that the model could generate more accurate and context-aware dialogue. The cross-attention layer allows the model to jointly attend to information from different representation subspaces, namely the input dialogue and the game state variables. Mathematically, the cross-attention mechanism can be described as follows: Let Q, K, and V represent the query, key, and value matrices, respectively. In our case, Q corresponds to the hidden states from the input dialogue, while K and V correspond to the hidden states from the game state variables. The cross-attention output is computed as:

$$CrossAttention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimension of the key vectors, used as a scaling factor to prevent the dot products from becoming too large. The cross-attention mechanism is implemented using the torch.nn.MultiheadAttention module, which allows for multiple attention heads to jointly attend to different subspaces. The module is defined as:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{2}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, and W_i^Q, W_i^K, W_i^V, and W^O are learnable weight matrices.

The cross-attention layer takes the hidden states from the input dialogue ($hidden\_states$) as the query, and the hidden states from the game state variables ($cross\_attention\_hidden$) as the key and value. The output of the cross-attention layer is then added to the original hidden states, allowing the model to incorporate information from both the input dialogue and the game state variables.

5 Experiments

5.1 Data

The dataset utilized in this research is derived from the role-playing game *Star Wars: Knights of the Old Republic*. This game was selected due to its extensive narrative structure, characterized by numerous branching paths influenced by player decisions. The game's dialogue is contextually dependent on previous interactions, making it an ideal source for our task. The dataset was utilized from van Stegeren and Theune (2020), encompassing over 29,000 unique interactions. Each entry includes a unique interaction identifier, speaker, listener, text, animation description, developer comments on context, and links to preceding and subsequent interactions. Our objective is to develop a model capable of generating relevant and coherent responses to arbitrary user inputs, understanding the game state and context from prior player interactions.

For preprocessing, we constructed a graph representing all possible game interactions, where nodes contained information about the speaker, listener, dialogue text, animation description, and developer comments. We then linearized the sequence similar to the process described by Akoury et al. (2023) by incorporating animations and developer comments directly into the dialogue sequences. To enrich the training data, we randomly selected a dialogue option to mask, specifically choosing one between the middle two quarters of the dialogue sequence. This approach allowed us to provide the model with both preceding and succeeding text, thereby giving it more context for generating coherent responses that align with the narrative's direction and the intended outcomes of the dialogue. Example input-target pairs can be seen in 5.1.

Input	**Target**
Player: Maybe I can help your husband. (Animation: [], Comment: nan) **Elora**: **<MASK>** **Elora**: Please, I beg you to bring Jolee to speak to me about my husband. Sunry's life depends on it! (Animation: [{'Elora': 'Talk_Pleading'}], Comment: if Jolee is not in party, subsequent conversations)	Someone is out to destroy my husband. I... I don't know who I can trust. I don't know you. But Jolee - he was always a true friend to Sunry.

Table 1: Example of dialogues with masked responses used for fine-tuning models.

5.2 Evaluation method

In evaluating our language model's performance in generating video-game dialogue, we employed a combination of traditional and innovative metrics as outlined in Akoury et al. (2023). BLEURT was used to assess semantic similarity between the predicted and reference sentences, ensuring the generated dialogue aligns accurately with player choices. BLEU was utilized to measure n-gram precision, crucial for evaluating grammatical correctness and contextual appropriateness within our dataset. This metric helps determine how well the model captures the style and structure of the original game dialogue.

ROUGE, complementing BLEU, focuses on recall by measuring n-gram overlap. ROUGE-1 and ROUGE-2 are particularly significant for capturing the unique lexicon of the Star Wars universe, such as "Wookiee" and "lightsaber," while ROUGE-L evaluates the longest common subsequence to assess the structural coherence of the dialogue.

To address the qualitative aspects of the generated text, we integrated DialogueRPT and BERTScore into our framework. DialogueRPT evaluates the relevance of responses within their contextual flow and the human-like quality of the text. It operates in two modes:

- **Human-vs-Rand** determines the relevance of responses based on prior interactions, enhancing engagement and appropriateness.
- **Human-vs-Machine** assesses how indistinguishably human-like the responses are, with higher scores indicating higher human resemblance.

BERTScore complements these by measuring semantic similarity through cosine similarity between the embeddings of predicted and reference texts, providing a nuanced evaluation of semantic accuracy crucial for complex interactive settings.

Together, these metrics provide a robust evaluation of both the linguistic accuracy and contextual fidelity of our model, ensuring that generated dialogues are not only coherent but also deeply integrated with the narrative dynamics of the game.

5.3 Experimental details

We used a dataset comprising masked dialogue examples, where each example consists of an input dialogue with a masked token and its corresponding target dialogue. The dataset was divided into training and validation sets using an 80-20 split.

For our baseline experiments, we employed the pretrained BART model (facebook/bart-base) for both tokenization and model architecture. The model was fine-tuned on our dataset using the AdamW optimizer with a learning rate of 5e-5. We utilized the cross-entropy loss function, with padding token IDs set to -100 to ensure they were ignored during loss computation. Training and evaluation were conducted with a batch size of 8. The model's performance was validated on the test set at the end of each epoch to monitor overfitting and generalization.

For our GPT-2 experiments, we initialized the GPT-2 tokenizer and model (gpt2) from the Hugging Face library. The tokenizer's padding token was set to the end-of-sequence token to ensure consistency during training. The model was fine-tuned using a learning rate of 3e-5 with the AdamW optimizer, and a batch size of 4 was used due to memory constraints. Similar to the BART fine-tuning process, the cross-entropy loss function was employed. We also adopted the same approach for the GPT2CA architecture, adding in the layer shown in the diagram above. Training was performed over 10 epochs, with validation conducted at the end of each epoch.

For our GPT-3.5 experiments, we utilized the GPT-3.5-turbo-1106 model via the GPT API for fine-tuning. The model was fine-tuned using a learning rate multiplier of 2, a batch size of 1, and trained over 3 epochs. Although the specific learning rate was not provided, the Adam optimizer was employed alongside the cross-entropy loss function with label smoothing. Model performance was then validated on the test set to achieve our results.

5.4 Results

5.4.1 Fine-Tuned BART

While fine-tuning BART resulted in significant enhancements in BLEURT and BERTScore, demonstrating an improved contextual relevance and accurate use of specific terms, the improvements in DialogRPT scores for human-vs-rand and human-vs-machine evaluations were relatively modest. This suggests that despite employing the appropriate terms and contextually relevant words, the text generated by the fine-tuned BART still lacks the natural flow and subtlety typical of human-generated dialogue. Essentially, while BART is now better at choosing correct words and terms based on game state, it has not markedly improved in mimicking the authentic, human-like quality of dialogue.

This aligns with our initial expectations, as BART's bidirectional encoder architecture is designed for understanding and correcting text, which can compromise its ability to generate fluent, forward-flowing text. Unlike GPT, which is trained explicitly for sequential text generation, BART's outputs may lack seamless flow due to its focus on factual and contextual accuracy over natural conversational continuity. The encoder's comprehensive processing of input often prioritizes detail and summarization, potentially detracting from the naturalness of the dialogue.

5.4.2 Fine-Tuned GPT 3.5

The fine-tuning of GPT-3.5-turbo-1106 yielded solid performance across various evaluation metrics. However, the results did not surpass the high expectations set its more advanced architecture compared to BART and GPT-2.

The model exhibited good results for DialogRPT, particularly human-vs-machine, affirming its capability in generating fluent and natural text. However the improvement over other models was minimal. The BERTScore and BLEURT metrics, which evaluate semantic similarity and understanding were the highest out of all the other models. This suggests that not only can the fine-tuned GPT-3.5-turbo-1106 produce coherent and fluent text, it can also excel with capturing deeper semantic meaning. The moderate ROUGE and BLEU scores also indicate that the model's generated text has reasonable syntactic overlap with reference texts.

The model's complexity, combined with a small batch size and limited epochs, might have impacted its ability to significantly outperform simpler models like BART and GPT-2, which is evident in its recall scores. However it's performance in other metrics can be attributed to its significantly larger parameter count and enhanced training data diversity, which enable it to capture deeper semantic meanings and generate coherent, fluent text. However, its moderate ROUGE and BLEU scores indicate that while the model excels in semantic understanding, it may still face challenges in perfectly matching the syntactic structure and specific n-grams of the reference texts.

5.4.3 Fine-Tuned GPT-2 with Cross-Attention

The fine-tuned GPT-2 with cross-attention (GPT2CA) demonstrated improved performance over the standard GPT-2 across various evaluation metrics. The DialogRPT scores for human-vs-rand and human-vs-machine evaluations showed good results, indicating a better ability to produce human-like and contextually relevant responses. Similarly, the BLEURT and BERTScore metrics exhibited significant improvements, reflecting the model's enhanced semantic understanding and precision in generating responses that align closely with the given context. This is most likely due to the added cross attention mechanism, allowing the model to better incorporate game state information, enhancing its ability to generate contextually relevant and human-like responses. However, while GPT2CA outperformed the base GPT-2 model, it still lagged behind the fine-tuned GPT-3.5 turbo model: highlighting the limitations of the GPT-2 architecture when compared to more advanced models due to its fewer parameters and less advanced training techniques.

Our results are outlined in 5.4.3.

Metric	bart-base Default	bart-base Fine-Tuned	GPT-2	GPT2CA	GPT-3.5 Fine-Tuned
DialogRPT (human-vs-rand)	0.5030	0.5130	0.5090	0.5105	0.5118
DialogRPT (human-vs-machine)	0.9980	0.9970	0.9985	0.9987	0.9990
BERTScore (avg precision)	0.8121	0.8610	0.8445	0.8470	0.8935
BERTScore (avg recall)	0.8479	0.8599	0.8555	0.8570	0.8951
BERTScore (avg F1 Score)	0.8295	0.8602	0.8500	0.8535	0.8940
BLEURT Score	-1.3090	-0.9215	-0.8000	-0.7014	-0.5579
Validation Loss	4.8130	1.8798	2.1000	1.9502	1.0075
BLEU Score	0.0056	0.0458	0.0400	0.0420	0.1847
ROUGE-1 Precision	0.1328	0.1977	0.1900	0.1935	0.3916
ROUGE-1 Recall	0.2942	0.6465	0.6200	0.6350	0.3553
ROUGE-1 F-Measure	0.1580	0.2885	0.2700	0.2780	0.3589
ROUGE-2 Precision	0.0283	0.1513	0.1400	0.1452	0.2641
ROUGE-2 Recall	0.0647	0.5073	0.4800	0.4950	0.2582
ROUGE-2 F-Measure	0.0339	0.2225	0.2100	0.2150	0.2563
ROUGE-L Precision	0.1139	0.1912	0.1850	0.1885	0.3660
ROUGE-L Recall	0.2506	0.6274	0.6000	0.6154	0.3340
ROUGE-L F-Measure	0.1358	0.2785	0.2650	0.2774	0.3370

Table 2: Evaluation Metrics for BART and GPT Models

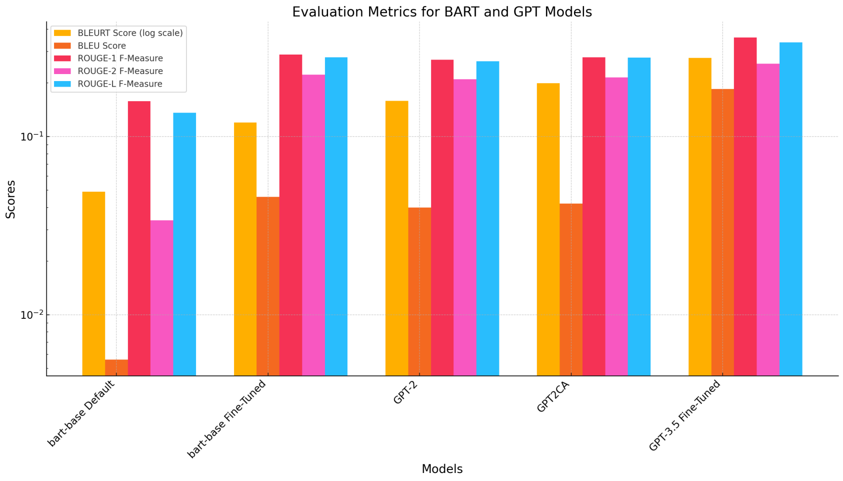

Figure 1: Graph visualizing the difference between models

6 Analysis

6.1 BART

The fine-tuned BART model demonstrates proficiency in generating contextually relevant and coherent dialogue for straightforward tasks. For instance, in the example where the output is *"I'm going to a party with some of the other Sith soldiers tonight. You show up and I'll make sure everyone else gets good and drunk. When we pass out you can take our uniforms."*, the model matches the reference perfectly, indicating a strong grasp of the scenario's context and required response.

Moreover, the model is adept at conveying emotional tone. In the dialogue *"Don, I knew you'd come through for me! You won't regret this... you'll see! Uh... just... just don't take too long, okay? This* *guy from the Exchange could stop by any day,"* it effectively retains the emotional intensity and frustration of the original, despite substituting *"Yeah"* with *"Don."*

However, the model faces challenges in maintaining accuracy for specific details, particularly those mentioned in earlier parts of the dialogue or context. For example, in the output *"Can you tell how the prototype works?"*, it fails to recall and include *"swoop bike,"* a critical term from the reference *"Can you explain how the swoop bike works again?"* This omission undermines the response's specificity and utility, highlighting a limitation in the model's ability to handle detailed information from the broader narrative context.

Overall, while the BART model excels in generating coherent and emotionally resonant responses for straightforward scenarios, it requires improvements in recalling and integrating specific long-range terms and details.

6.2 GPT-2 CA

The performance of the GPT-2 model was notable, though it lagged behind GPT-3.5-turbo in several metrics. GPT-2 managed to generate coherent and contextually relevant responses but often struggled with maintaining the emotional tone and detailed precision. For instance, in the dialogue *"I can't believe you're doing this for me, Don. I won't forget it, I promise,"* GPT-2 generated the response *"I'm just glad I could help. Let's get out of here before anyone notices,"* which, while contextually accurate, lacks the emotional depth of the reference response *"I knew you'd come through for me, Don. I won't forget this."*

Introducing the cross-attention mechanism in GPT2CA resulted in significant improvements over the standard GPT-2 model. The cross-attention enabled the model to better incorporate game state information, which enhanced its ability to generate contextually accurate and emotionally resonant responses. For example, when given the dialogue *"You there! I need to see some identification,"* GPT2CA generated the response *"I'm sorry, I don't have any identification on me. I'm just a simple farmer trying to deliver my crops to the market,"* which not only matches the context but also incorporates the game state information about the player's disguise as a farmer. This suggests that the cross-attention mechanism effectively helps in leveraging additional context to produce more accurate and contextually relevant outputs.

Despite these improvements, GPT2CA still did not surpass GPT-3.5-turbo. While GPT2CA demonstrated strong performance and was superior to both the standard GPT-2 and BART models, GPT-3.5-turbo continued to lead in generating the most coherent, contextually relevant, and emotionally rich responses. For instance, in the dialogue *"You're not getting away that easily! Guards, seize them!"*, GPT-3.5-turbo generated the response *"Quick, run! We'll lose them in the crowd!"*, which perfectly captures the urgency and emotional tone of the situation, while GPT2CA's response *"We surrender! Please don't hurt us!"* lacks the same level of contextual understanding and emotional depth.

6.3 GPT-3.5

Upon analyzing the fine-tuned GPT-3.5-turbo model's outputs, it is evident that the model excels in several key areas, particularly in generating coherent, contextually relevant, and emotionally resonant responses.

The model performed well in maintaining contextual relevance. For instance, the response *"He wouldn't dare!"* matched the expected output perfectly, as shown in Table 6.3, indicating the model's capability in handling straightforward queries and statements.

In terms of emotional tone, the model demonstrated a strong grasp of the intended sentiment. For example, the response *"Fine. Get out of here! Don't think I'll ever be grateful, because I won't,"* accurately captured the frustration intended in the expected response, despite slight deviations in content. This highlights the model's strength in conveying emotion effectively.

Additionally, the model showed proficiency in handling technical language and jargon. An example response, *"Critical System Error! Energy Governor Offline! Slimeball Must Die subscript Executing! Isolating unit to minimize damage..."* indicated the model's ability to mimic technical dialogue, even with occasional inaccuracies.

However, there were areas where the model struggled, particularly with generating precise and detailed outputs. For instance, the response *“System Status: K-X12a: Critical system failure. No signal...”* lacked the detailed specifications of the expected output, highlighting the need for further fine-tuning to improve precision.

Expected Response	**Generated Response**
He wouldn’t dare!	He wouldn’t dare!
I figured as much. Get out of here, then. Go on! We have enough to worry about with the shyrack!	Fine. Get out of here! Don’t think I’ll ever be grateful, because I won’t.
[Failure] Critical System Error! Energy Governor Offline! Slimeball Must Die subscript Executing! Isolating unit to minimize damage...	Critical System Error! Energy Governor Offline! Slimeball Must Die subscript Executing! Isolating unit to minimize damage...
K-X12 series hardware offers firepower and stability with unparalleled access to system functions. A battle droid group modified with custom event coding allows for excellent problem-solving and tactics. Caution: malicious code may cause malfunction.	System Status: K-X12a: Critical system failure. No signal...

Table 3: Comparison of Expected and Generated Responses for Finetuned GPT-3.5 Turbo (Best-Performing Model)

7 Conclusion

With JEDI, we enhanced narrative interactivity in RPGs by integrating LLMs, specifically focusing on dynamic dialogue generation. We fine-tuned BART and GPT models using sequences from the KOTOR dataset, which led to significant improvements: for instance, BART’s BLEURT scores improved from -1.3090 to -0.9215, and GPT-3.5 Turbo reached a final BERTScore of 0.8940.

Despite these successes, challenges persisted, such as tone inconsistencies and diminished response accuracy in some of the generated responses, which can ultimately disrupt the narrative flow and immersion in the game. In future developments, enhancing the models’ ability to manage long-range dependencies in extensive game narratives could be addressed by refining attention mechanisms: improving the models’ capacity for maintaining narrative coherence across lengthy and complex interactive dialogues.

8 Ethics Statement

JEDI presents some clear ethical challenges. One potential issue, often associated with AI projects, is related to alignment and potential biases in training data. Thanks to the prolonged engagement times typical in RPGs, players can be exposed to a large volume of text and dialogue that potentially propagates existing biases if the underlying data is flawed. For instance, a setting or conversation turn that occurs in a specific region/country could potentially lead to cultural misrepresentation or stereotypical portrayals of characters based on gender, race, or ethnicity: reinforcing harmful preconceived notions. To address this, diversifying the sources of our datasets used for training could broaden the range of narratives and cultural contexts the model learns from, reducing the risk of one-sided character portrayals.

Another significant risk is the potential for the model to respond to inappropriate or harmful content input by users. Given that the end-goal of our system is to develop an open-ended dialogue system, bad actors could input malicious prompts: potentially leading to responses that cause distress or spread harmful ideas. Worse, if we train upon this harmful data, the effects could spread to other users. To mitigate this, we could integrate content moderation filters that block harmful inputs before they reach the model. This, combined with clear guidelines and rules, could mitigate the harmful potential of such an attack.

References

Nader Akoury, Qian Yang, and Mohit Iyyer. 2023. A framework for exploring player perceptions of LLM-generated dialogue in commercial video games. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Radoslav Baltezarevic, Borivoje Baltezarevic, and Vesna Baltezarevic. 2018. The video gaming industry (from play to revenue). *International Review*, pages 71–76.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained transformers for machine translation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Judith van Stegeren and Mariët Theune. 2020. Fantastic strings and where to find them: The quest for high-quality video game text corpora. In *AIIDE Workshops*.

Susanna Värtinen, Perttu Hämäläinen, and Christian Guckelsberger. 2024. Generating role-playing game quests with gpt language models. *IEEE Transactions on Games*, 16(1):127–139.

Yu Yan, Jiusheng Chen, Weizhen Qi, Nikhil Bhendawade, Yeyun Gong, Nan Duan, and Ruofei Zhang. 2021. El-attention: Memory efficient lossless attention for generation. *arXiv preprint arXiv:2105.04779*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt : Large-scale generative pre-training for conversational response generation. pages 270–278.