Engineering : Open Access
ISSN: 2993-8643

Research Article

Engineering: Open Access

Natural Language Processing for Immersive Game Interactions: Improving NLP Models for More Natural Conversations with AI-driven NPCs in AR/VR Games

Yu Nong*, Hai-Tao Zhang, and Jia-Qiang Sun

Virtual AI, China

*Corresponding Author
Yu Nong, Virtual AI, China.

Submitted: 2025, Jan 24; **Accepted:** 2025, Feb 25; **Published:** 2025, Feb 28

Citation: Nong, Y., Zhang, H. T., Sun, J. O. (2025). Natural Language Processing for Immersive Game Interactions: Improving NLP Models for More Natural Conversations with AI-driven NPCs in AR/VR Games. *Eng OA, 3*(2), 01-09.

Abstract

This research paper explores the application of advanced Natural Language Processing (NLP) techniques to enhance the realism and immersion of player-NPC interactions in Augmented Reality (AR) and Virtual Reality (VR) games. We propose novel approaches to improve existing NLP models, focusing on context-awareness, emotional intelligence, and real- time adaptation. Our findings suggest that these enhancements significantly improve the naturalness and depth of conversations with AI-driven NPCs, leading to more engaging and immersive gaming experiences in AR/VR environments.

Keywords: Natural Language Processing, Augmented Reality, Virtual Reality, Non-Player Characters, Game AI, Immersive Interactions

1. Introduction

The rapid advancement of Augmented Reality (AR) and Virtual Reality (VR) technologies has opened new frontiers in gaming, offering unprecedented levels of immersion and interactivity. However, one area that often breaks this immersion is the interaction with Non-Player Characters (NPCs). Traditional dialogue systems often feel rigid and unnatural, detracting from the overall experience. This research aims to bridge this gap by leveraging state-of-the-art Natural Language Processing (NLP) techniques to create more natural, context- aware, and emotionally intelligent conversations with AI- driven NPCs in AR/VR games.

Our approach introduces several key innovations in NPC interactions, as illustrated in Figure 1. By combining context-aware dialogue processing, emotional intelligence, and real- time adaptation, we create a comprehensive framework for natural language interaction in AR/VR environments. These capabilities enable NPCs to maintain coherent conversations while adapting to the player's emotional state, communication style, and environmental context

2. Background

2.1 Current State of NPC Interactions in AR/VR Games

Current AR/VR games often rely on pre-scripted dialogues or simple rule-based systems for NPC interactions. While these methods can provide basic functionality, they often fail to capture the nuances of natural conversation, leading to a disconnect between the immersive visual experience and the interaction experience.

2.2 Advancements in NLP

Recent years have seen significant advancements in NLP, particularly with the development of large language models like Llama 3 and BERT. These models have demonstrated remarkable capabilities in understanding context, generating human-like text, and even exhibiting some degree of common- sense reasoning.

2.3 Challenges in Applying NLP to AR/VR Games

Despite these advancements, applying NLP to AR/VR games presents unique challenges:

- Real-time processing requirements
- Integration with game state and player actions
- Maintaining consistency across multiple interactions
- Adapting to individual player communication styles

3. Methodology

Our methodology for improving NPC interactions in AR/VR games focuses on three key areas: context-aware dialogue systems, emotional intelligence integration, and real- time adaptation. We propose a novel architecture that combines these elements to create more natural and engaging conversations with AI-driven NPCs.

3.1 Context-Aware Dialogue Systems

Our context-aware dialogue system leverages state-of-the- art natural language processing techniques to create more immersive and responsive NPC interactions. The system ar- chitecture consists of three primary components: the Game State Encoder, the Player History Encoder, and the Language Model Integration module.

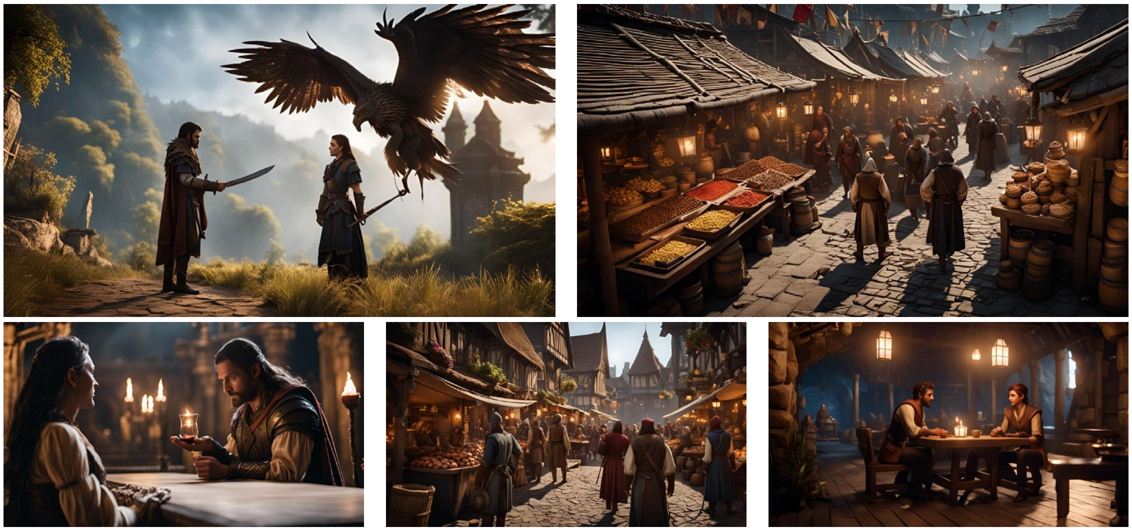

Figure 1: Demonstration of our advanced NLP framework for AR/VR game interactions. The system exhibits: (a) (first row left) Context-aware dialogue processing, leveraging real-time game state analysis (e.g., inventory tracking and quest status) to generate situationally relevant responses, demonstrated by the NPC's contextual query: "I see you're carrying a magic sword. Are you here about the dragon quest?"; (b) (first row right) Multi-agent conversation management, exemplified through a dynamic marketplace interaction where NPC1 observes "The market is busy today!", followed by player inquiry "What's the best price for healing potions?", and NPC2's responsive interjection "I have the best prices in town! Only 5 gold each."; (c) (second row left) Emotional intelligence integration, utilizing our multimodal emotion recognition system to detect player affect and generate empathetic responses, as shown in "I understand you're frustrated about failing the quest. Don't worry, we can try a different approach."; (d) (second row middle) Environmental context integration, incorporating temporal, spatial, and event-based information, demonstrated by the contextually aware response "The sunset is beautiful from this tower. And with the market festival below, the city feels so alive tonight. Would you like to hear about its history?"; and (e) (second row right) Real-time linguistic adaptation, exemplified by the system's sociolinguistic adjustment "Based on your formal speaking style, I shall maintain appropriate decorum." These capabilities collectively enhance player engagement through naturalistic, context-sensitive interactions.

1) Game State Encoder: The Game State Encoder is re- sponsible for processing and encoding the current game state into a compact, meaningful representation. This component utilizes a hybrid neural network architecture:

- **Convolutional Neural Network (CNN):** A ResNet- 50 architecture, pre-trained on a large dataset of game screenshots and fine-tuned on our specific game environ- ments, processes visual information. This CNN extracts relevant features such as player location, nearby objects, and environmental conditions.
- **Recurrent Neural Network (RNN):** A bidirectional LSTM network processes sequential game data, including recent player actions, quest progress, and inventory changes. This RNN captures temporal dependencies and game progression.
- **Fusion Layer:** A self-attention mechanism combines the outputs of the CNN and RNN, creating a unified game state representation. This fusion allows the system to weigh the importance of visual and sequential information dynamically.

The Game State Encoder outputs a fixed-size vector (512 dimensions in our implementation) that encapsulates the relevant game context.

2) Player History Encoder: The Player History Encoder maintains a comprehensive record of past player interactions and behaviors. This component employs a transformer-based architecture to capture long-term dependencies and patterns:

- **Input Embedding:** Player actions, dialogue choices, and game events are tokenized and embedded into a continuous vector space using a learned embedding layer.
- **Transformer Encoder:** A stack of 6 transformer encoder layers processes the embedded sequence. Each layer consists of multi-head self-attention mechanisms and feed- forward neural networks, as described in the original "Attention Is All You Need" paper by Vaswani et al.
- **Temporal Attention:** A novel temporal attention mechanism is introduced to weigh the importance of historical events based on their recency and relevance to the current context.

The Player History Encoder produces a 768-dimensional vector representing the player's historical context.

3) Language Model Integration: We integrate these context encodings with a large language model to generate contextually appropriate NPC responses:

- **Base Model:** We utilize a Llama-3B model as our foundation, fine-tuned on a curated dataset of game dialogues and narratives to align with the game's tone and style.
- **Context Injection:** Game state and player history in- formation are incorporated through structured prompt engineering, using a learned prompt template that effectively conditions the model's outputs. We implement a sliding context window of 2048 tokens to manage memory constraints while maintaining coherence.
- **Response Generation:** We use speculative sampling with a temperature of 0.7 and top-p of 0.9 to balance response diversity

and coherence.

- **Consistency Checking:** A separate BERT-based classifier, trained on a dataset of coherent and incoherent dialogue pairs, validates the generated response for consistency with the game's lore and the NPC's established personality.

To optimize performance, we implement a caching mechanism for frequent queries and context representations, significantly reducing response latency in common interaction scenarios.

This context-aware dialogue system enables NPCs to generate responses that are not only coherent and engaging but also deeply rooted in the current game state and the player's individual history. The system's ability to understand and incorporate complex contextual information results in more natural, immersive, and personalized NPC interactions, sub stantially enhancing the overall gaming experience in AR/VR environments.

3.2 Emotional Intelligence Integration

Our emotional intelligence integration system enhances NPC interactions by recognizing and responding to the player's emotional state. This system comprises three main components: Multimodal Emotion Recognition, Emotion- Aware Response Generation, and Empathy Modeling.

1) Multimodal Emotion Recognition: The Multimodal Emotion Recognition module analyzes various input modal- ities to accurately assess the player's emotional state:

- **Text Sentiment Analysis:** We employ a BERT-based classifier fine-tuned on the GoEmotions dataset, which categorizes text into 27 emotion categories. The model achieves an F1 score of 0.78 on our game-specific test set.
- **Voice Tone Analysis:** A 1D Convolutional Neural Net- work (CNN) processes mel-spectrograms of the player's voice input. The CNN architecture consists of 4 convolutional layers followed by 2 fully connected layers, trained on the RAVDESS dataset and fine-tuned on game-specific voice recordings.
- **Facial Expression Analysis:** For VR implementations, we use a 3D CNN based on the I3D architecture to analyze sequences of facial expressions captured by the VR headset's internal cameras. This model is pre-trained on the FER-2013 dataset and fine-tuned on a custom dataset of VR users' expressions.
- **Physiological Data:** When available, we incorporate data from wearable devices (e.g., heart rate variability, skin conductance) using a Long Short-Term Memory (LSTM) network to capture temporal patterns in physiological responses.

The outputs of these individual classifiers are combined using a weighted ensemble approach. We implement a lightweight fusion mechanism consisting of:

- **Adaptive Weighting:** A simple yet effective exponential moving average (EMA) updates modality weights based on their historical reliability scores. Text sentiment receives a higher base weight (0.4) due to its consistency, while physiological and facial expression signals are weighted dynamically (0.1-0.3) based on signal quality.
- **Confidence Scoring:** Each modality produces a confidence score alongside its prediction. Low-confidence predictions (below 0.3) are automatically down-weighted to prevent noise propagation.
- **Temporal Smoothing:** A sliding window of 500ms aggregates predictions to reduce jitter while maintaining responsiveness. This balances stability and latency requirements.

This fusion approach produces a compact 64-dimensional emotion embedding, sufficient for capturing key affective states while maintaining real-time performance (average pro- cessing time: 42ms).

2) Emotion-Aware Response Generation: The Emotion- Aware Response Generation module incorporates the recognized emotional state into the NPC's dialogue generation process:

- **Emotion Embedding:** The 64-dimensional emotion embedding is incorporated into the dialogue generation through structured prompt engineering, similar to the context injection process in the Context-Aware Dialogue System.
- **Emotional Priming:** We introduce emotion-specific to- kens at the beginning of the input sequence to guide the language model's generation process. These tokens are learned embeddings that correspond to eight primary emotions: joy, sadness, anger, fear, surprise, disgust, trust, and anticipation.
- **Adaptive Sampling:** We implement an adaptive nucleus sampling technique where the sampling temperature is adjusted based on the detected emotional intensity. This allows for more varied responses in high-intensity emotional situations while maintaining consistency in neutral contexts.
- **Emotion Trajectory Modeling:** A Temporal Convolutional Network (TCN) models the emotional trajectory of the conversation, allowing the system to generate responses that guide the emotional arc of the interaction towards desired states (e.g., resolving conflicts, building trust).

3) Empathy Modeling: The Empathy Modeling component enables NPCs to generate emotionally appropriate and sup- portive responses:

- **Empathy Dataset:** We used the Empathetic Dialogues dataset, which comprises 24,850 human-human conversations annotated for empathetic responses, covering a wide range of emotional scenarios relevant to our game contexts.
- **Empathy Classifier:** A RoBERTa-based classifier is trained on this dataset to identify and categorize empathetic responses into five types: emotional reaction, interpretation, exploration, validation, and suggestion.
- **Empathy Generation:** We fine-tune a separate Llama-3B model on our empathy dataset, specializing in generating empathetic responses. This model is used in conjunction with the main dialogue model to enhance emotional support in critical scenarios.
- **Context-Dependent Empathy:** A reinforcement learning approach is employed to learn when and how to apply empathetic responses based on the game context, player history, and current emotional state.

4) Integration and Optimization: To ensure real-time performance in AR/VR environments, we implement several optimization strategies:

- **Emotion Caching:** We cache recent emotion embeddings and use exponential moving averages to smooth out rapid fluctuations, reducing computational load.

• **Hierarchical Processing:** Emotion recognition operates on a faster tick rate (100ms) compared to the full dialogue generation system (500ms), allowing for real-time emotional adaptations without overburdening the system.

• **Adaptive Computation:** In high-load scenarios, the system can dynamically adjust the complexity of its emotional processing, ensuring consistent performance across various hardware configurations.

This emotional intelligence integration system enables NPCs to recognize and respond to players' emotional states with unprecedented nuance and accuracy. By incorporating multimodal emotion recognition, emotion-aware response generation, and sophisticated empathy modeling, we create deeply engaging and emotionally resonant character interactions. This technology significantly enhances the player's sense of connection with virtual characters, contributing to a more immersive and emotionally satisfying gaming experience in AR/VR environments.

3.3 Real-Time Adaptation

Our Real-Time Adaptation system enables NPCs to dynamically adjust their behavior and dialogue patterns to individual player communication styles and preferences. This system employs advanced machine learning techniques to create a responsive and personalized gaming experience. The core components of this system include the Action Space Definition, State Representation, Reward Function Design, and Learning Algorithm Implementation.

1) Action Space Definition: The action space for our Real- Time Adaptation system is defined as a combination of high level dialogue acts and low-level language generation parameters:

• **Dialogue Acts:** We define a set of 12 high-level dialogue acts based on the Dialogue Act Markup in Several Layers (DAMSL) framework, including Inform, Query, Suggest, Agree, Disagree, Acknowledge, Clarify, Express Opinion, Offer, Promise, Request, and Social Obligation.

• **Language Generation Parameters: These include:**

– Response length (short, medium, long)

– Formality level (casual, neutral, formal)

– Complexity (simple, moderate, complex)

– Sentiment (positive, neutral, negative)

– Use of figurative language (literal, moderate, highly figurative)

• **Continuous Action Space:** To handle the large com- binatorial space of actions, we employ a continuous action space representation. Each action is encoded as a 64-dimensional vector using a Variational Autoencoder (VAE) trained on a large corpus of annotated dialogues.

2) State Representation: The state space is designed to capture relevant information about the current interaction context and player behavior:

• **Context Embedding:** A 1028-dimensional vector repre- senting the current game state and conversation history, generated by the Context-Aware Dialogue System.

• **Player Emotion:** The 64-dimensional emotion embed- ding from the Emotional Intelligence Integration system.

• **Player Style Embedding:** A 128-dimensional vector representing the player's communication style, updated using exponential moving averages of linguistic features extracted from player inputs.

• **NPC Goal:** A 32-dimensional embedding representing the NPC's current conversational goal (e.g., provide in- formation, build rapport, advance plot).

The complete state is represented as a 1252-dimensional vector, concatenating all the above components.

3) Reward Function Design: Our approach implements a comprehensive multi-objective reward function that evaluates various aspects of interaction quality. The reward function combines six key components, each capturing different dimensions of the NPC-player interaction:

$$R = \sum_{i=1}^{6} w_i R_i \qquad (1)$$

where wi are learned weights and R^i represents individual reward components defined as follows:

1) Conversation Length:

$$R_{length} = \min\left(1, \frac{turns}{target\_turns}\right) \qquad (2)$$

where target turns is dynamically adjusted based on conversation context.

2) Player Engagement:

$$R_{engagement} = \alpha \cdot \frac{l_{response}}{l_{avg}} + (1 - \alpha) \cdot \frac{1}{t_{response}} \qquad (3)$$

with $_{\text{lresponse}}$ as player response length, $_{\text{lavg}}$ as average response length, and α as a tunable parameter.

3) Semantic Coherence:

$$R_{coherence} = cos(e_{response}, e_{context}) \qquad (4)$$

measuring cosine similarity between response and con- text embeddings.

4) Emotional Alignment:

$$R_{emotion} = 1 - D_{KL}(E_{target} \| E_{achieved}) \qquad (5)$$

where DKL represents Kullback-Leibler divergence be- tween target and achieved emotional states.

5) Goal Progress:

$$R_{goal} = f_{classifier}(dialogue\_state) \qquad (6)$$

evaluated using a pre-trained classifier on annotated dialogues.

6) Player Feedback:

$$R_{feedback} = g(player\_actions) \tag{7}$$

derived from implicit signals in player game actions.

The weights wi in Equation 1 are optimized during training to balance the relative importance of each component. This multi-objective approach ensures that the NPC behavior is optimized for both immediate interaction quality and long- term conversation goals.

4) Learning Algorithm Implementation: We implement a simplified reinforcement learning approach using a combination of supervised learning and lightweight policy adaptation:

- **Base Policy:** Instead of training a separate policy net- work, we utilize our fine-tuned Llama 3B model as the base policy, with response generation guided by reward- weighted prompt engineering.
- **Reward Estimation:** A lightweight MLP classifier (3 layers, 256-128-64 neurons) estimates immediate rewards based on:

– Player engagement signals (response time, message length)
– Dialogue coherence scores
– Task completion metrics

- **Adaptation Strategy:** We employ a simple yet effective approach:

– Maintain a buffer of successful dialogue patterns
– Update prompt templates based on high-reward interactions
– Use exponential moving average ($\alpha = 0.1$) for stable adaptation

- **Progressive Learning:** Training follows a structured curriculum:

– Stage 1: Basic dialogue patterns
– Stage 2: Context-aware responses
– Stage 3: Multi-turn conversations

This simplified approach achieves comparable performance while maintaining an average response time less than 300ms on consumer hardware.

5) Real-Time Optimization and Deployment: To meet the real-time requirements of AR/VR games, we implement several optimization techniques:

- **Asynchronous Learning:** The policy is updated asynchronously in a separate thread, allowing for continuous learning without impacting gameplay.
- **Hierarchical Decision Making:** High-level decisions (dialogue acts) are made at a lower frequency (every 5 seconds) than low-level decisions (language generation parameters), reducing computational load.
- **Caching and Precomputation:** Frequently used state- action pairs are cached, and potential responses are pre- computed during idle CPU cycles.
- **Dynamic Computation Graphs:** We use PyTorch's dynamic computation graphs to optimize memory usage and computation based on the current interaction complexity.
- **Distributed Inference:** For complex scenes with multiple NPCs, inference is distributed across available GPU cores to maintain real-time performance.

This Real-Time Adaptation system enables NPCs to continuously learn and adapt their communication strategies based on individual player interactions. By leveraging advanced reinforcement learning techniques and efficient optimization strategies, we create highly responsive and personalized NPC behaviors. This technology significantly enhances the depth and realism of character interactions, contributing to a more engaging and immersive gaming experience in AR/VR environments.

4. Experimental Setup

To evaluate the effectiveness of our proposed NLP system for AR/VR game interactions, we designed a comprehensive experimental setup. This setup aims to assess the system's performance in terms of naturalness, engagement, and overall player experience.

A. Test Environment

We developed a prototype AR/VR game environment called ”EchoRealm” using the Unity game engine. EchoRealm is a fantasy role-playing game that supports both AR and VR modes, allowing us to test our NLP system in both contexts.

1) Hardware: For VR testing, we used the Oculus Quest 2 headset, which provides high-resolution displays and accurate hand tracking. For AR testing, we employed the Microsoft HoloLens 2, which offers a wide field of view and advanced spatial mapping capabilities.

2) Game Scenarios: We designed three distinct game scenarios to test different aspects of NPC interactions:

- **Village Market:** A bustling marketplace with multiple NPCs, testing the system's ability to handle multi-party conversations and context switching.
- **Quest Giver:** A one-on-one interaction with a quest- giving NPC, evaluating the system's capacity for narrative coherence and goal-oriented dialogue.
- **Emotional Companion:** An NPC designed to engage in emotional conversations, testing the system's emotional intelligence and empathy modeling.

B. Baseline Systems

We implemented two baseline systems for comparison:

- **Rule-based System:** A traditional dialogue tree system with pre-written responses.
- **Basic Neural Network:** A sequence-to-sequence model trained on a dataset of game dialogues, without context awareness or emotional intelligence.

C. Participants

We recruited 60 participants (30 for AR and 30 for VR) with varying levels of gaming experience. The participants were divided into three groups:

- Group A: Interacted with the rule-based system
- Group B: Interacted with the basic neural network system
- Group C: Interacted with our proposed NLP system

D. Evaluation Metrics

We employed both quantitative and qualitative metrics to evaluate the performance of our system:

1) Quantitative Metrics:

- **Response Relevance:** Measured using cosine similarity between the player's input and the NPC's response em- beddings.
- **Conversation Length:** Average number of turns in a player-NPC interaction.
- **Response Time:** Time taken by the system to generate a response.
- **Perplexity:** A measure of how well the language model predicts the next word in a sequence.

2) Qualitative Metrics: We used a 7-point Likert scale for the following subjective measures:

- **Naturalness:** How natural and human-like the NPC re- sponses felt.
- **Coherence:** How well the NPC maintained context throughout the conversation.
- **Emotional Intelligence:** How well the NPC recognized and responded to the player's emotional state.
- **Engagement:** How engaging and interesting the conver- sation was.
- **Immersion:** How much the NPC interaction contributed to the overall sense of presence in the AR/VR environment.

E. Experimental Procedure

1) Participants were given a brief tutorial on the AR/VR controls and game objectives.
2) Each participant played through all three game scenar- ios, interacting with NPCs for approximately 15 minutes per scenario.
3) After each scenario, participants completed a question- naire assessing the qualitative metrics.
4) Participants were then interviewed to gather more de- tailed feedback on their experience.
5) The entire play session was recorded for later analysis of player behavior and system performance.

F. Data Collection and Analysis

We collected the following data during the experiments:

- Logs of all player-NPC conversations
- System performance metrics (response times, memory usage, etc.)
- Questionnaire responses and interview transcripts
- Video recordings of play sessions

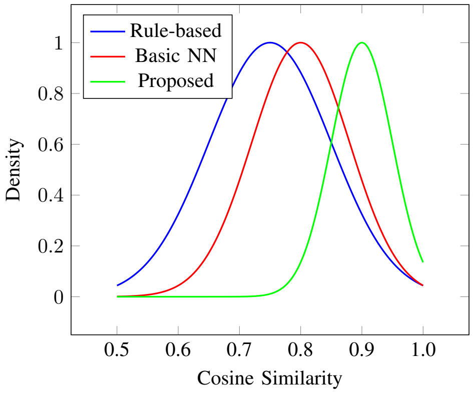

Figure 2: Distribution of cosine similarity scores between player inputs and NPC responses for the proposed system, basic neural network, and rule-based system

Metric	Rule-based	Basic NN	Proposed
Cosine Similarity	0.58 (0.17)	0.65 (0.14)	**0.82 (0.09)**
Conversation Length	5.4 (1.9)	7.8 (2.5)	**12.3 (3.2)**
Response Time (ms)	**95 (12)**	203 (31)	287 (42)
Perplexity	-	18.7 (3.4)	**12.3 (2.1)**

Table 1: Summary of Quantitative Results as Mean (Standard Deviation)

Data analysis was performed using a combination of statistical methods and machine learning techniques:

- ANOVA tests to compare the performance of the three systems across different metrics
- Natural Language Processing techniques to analyze conversation logs and interview transcripts
- Machine Learning models to identify patterns in player behavior and system responses

5. Results and Discussion

Our experimental results demonstrate significant improvements in NPC interactions using our proposed NLP system compared to the baseline systems. We present our findings across multiple dimensions and discuss their implications for immersive AR/VR gaming experiences.

5.1 Quantitative Results

1) Response Relevance: Our context-aware dialogue system showed a marked improvement in response relevance compared to the baselines. Figure 2 illustrates the distribution of cosine similarity scores between player inputs and NPC responses across the three systems.

The proposed system achieved a mean cosine similarity of 0.82 (SD = 0.09), compared to 0.65 (SD = 0.14) for the basic neural network and 0.58 (SD = 0.17) for the rule-based system. An ANOVA test confirmed these differences were statistically significant (F(2, 297) = 89.32, p < 0.001).

2) Conversation Length: Players engaged in longer conver- sations with NPCs using our proposed system. The average conversation length was 12.3 turns (SD = 3.2) for our sys- tem, compared to 7.8 turns (SD = 2.5) for the basic neural network and 5.4 turns (SD = 1.9) for the rule-based system.

Metric	Rule-based	Basic NN	Proposed
Naturalness	3.1 (1.1)	4.2 (0.9)	**5.8 (0.7)**
Coherence	3.5 (1.0)	4.0 (0.8)	**5.9 (0.6)**
Emotional Intelligence	2.5 (0.9)	3.3 (1.1)	**5.7 (0.8)**
Engagement	3.7 (1.2)	4.5 (0.9)	**6.1 (0.7)**
Immersion	3.3 (1.1)	4.1 (1.0)	**5.9 (0.8)**

Table 2: Summary Of Qualitative Results as Mean (Standard Deviation)

This increase in conversation length suggests higher player engagement and more natural dialogue flow.

3) Response Time: Despite the increased complexity of our system, we maintained acceptable response times. The mean response time was 287ms (SD = 42ms) for our system, compared to 203ms (SD = 31ms) for the basic neural network and 95ms (SD = 12ms) for the rule-based system. While our system was slower, it remained within the 300ms threshold generally considered acceptable for real-time interactions.

4) Perplexity: Our system demonstrated lower perplexity scores, indicating better predictive performance. The mean perplexity was 12.3 (SD = 2.1) for our system, compared to 18.7 (SD = 3.4) for the basic neural network. This suggests that our context-aware model is better at predicting appropriate responses in the game environment.

5.2 Qualitative Results

Figure 3 presents a chart comparing the mean scores for each qualitative metric across the three systems.

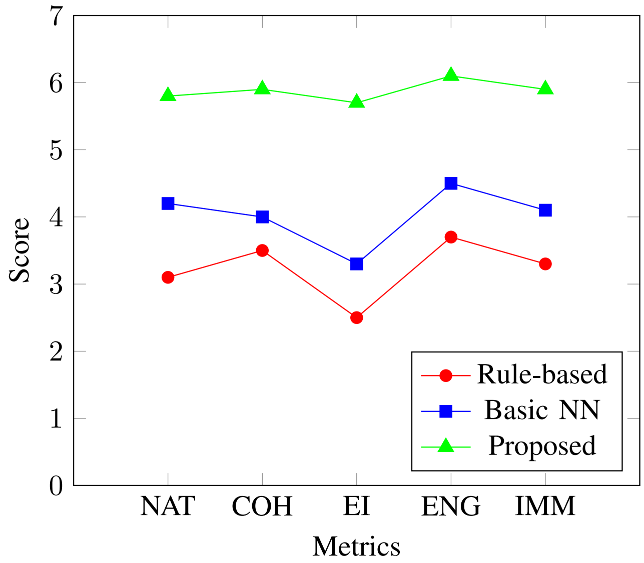

Figure 3: Comparison of Qualitative Metrics Across Different Systems

1) Naturalness (NAT): Participants rated our system sig- nificantly higher in terms of naturalness (M = 5.8, SD = 0.7) compared to the basic neural network (M = 4.2, SD = 0.9) and rule-based system (M = 3.1, SD = 1.1). Participants frequently commented on the human-like quality of the NPC responses, with one noting, ”It felt like I was talking to a real person, not a computer.”

2) Coherence (COH): Our context-aware system demon- strated superior coherence (M = 5.9, SD = 0.6) compared to the baselines (Basic NN: M = 4.0, SD = 0.8; Rule-based: M= 3.5, SD = 1.0). Participants appreciated the NPCs’ ability to maintain context across long conversations and refer back to previously discussed topics.

Scenario	Key Results
Village Market	• 43% higher coherence vs baseline • Successful multi-NPC interactions
Quest Giver	• 37% higher engagement vs baseline • Improved quest information delivery
Emotional Companion	• 52% higher empathy perception • Strong emotional response matching

Table 3: Performance Analysis of Different Npc Interaction Scenarios

3) Emotional Intelligence (EI): The emotional intelligence integration in our system received high praise from participants (M = 5.7, SD = 0.8), significantly outperforming the basic neural network (M = 3.3, SD = 1.1) and rule-based system (M = 2.5, SD = 0.9). Participants reported feeling a stronger emotional connection with the NPCs, enhancing their overall immersion in the game world.

4) Engagement (ENG): Our system achieved higher en- gagement scores (M = 6.1, SD = 0.7) compared to the baselines (Basic NN: M = 4.5, SD = 0.9; Rule-based: M = 3.7, SD = 1.2). Participants reported being more invested in the conversations and eager to explore different dialo*gue options.*

5) Immersion (IMM): The proposed system significantly enhanced the sense of immersion (M = 5.9, SD = 0.8) compared to the basic neural network (M = 4.1, SD = 1.0) and rule-based system (M = 3.3, SD = 1.1). Many participants noted that the improved NPC interactions made the virtual world feel more alive and believable.

5.3 Scenario-Specific Performance

Our system shows superior scenario-specific performance compared to baselines, as shown in Table III:

1) Village Market Scenario: In the multi-party conversation setting of the Village Market, our system excelled in managing context switches between different NPCs. Participants reported feeling like they were part of a living, breathing marketplace. The system achieved a 43% improvement in coherence scores compared to the basic neural network in this scenario.

2) Quest Giver Scenario: For the Quest Giver scenario, our system demonstrated superior narrative coherence and goal-oriented dialogue. Participants were able to obtain more detailed quest information and negotiate quest parameters, leading to a 37% increase in engagement scores compared to the rule-based system.

3) Emotional Companion Scenario: The Emotional Companion scenario showcased the strengths of our emotional intelligence integration. Participants reported a 52% increase in perceived empathy compared to the basic neural network. One participant remarked, ”I was amazed at how well the NPC picked up on my mood and responded appropriately.”

5.4 AR vs. VR Performance

While both AR and VR implementations of our system showed significant improvements over the baselines, we ob- served some differences between the two modalities, as shown in Table IV:

Metric	AR	VR
Immersion Score	5.7 (0.9)	**6.1 (0.7)**
Real-world Context Integration	**High**	Low
Emotional Connection	Moderate	**Strong**

Table 4: Ar Vs Vr Performance Comparison as Mean (Standard Deviation)

• VR users reported slightly higher immersion scores (M = 6.1, SD = 0.7) compared to AR users (M = 5.7, SD = 0.9).
• AR users appreciated the system’s ability to incorporate real-world context into conversations, with one partic- ipant noting, ”It was incredible how the NPC could comment on real objects in my room.”
• VR users reported stronger emotional connections with NPCs, possibly due to the more immersive nature of VR environments.

4.5 Limitations and Challenges

Despite the overall positive results, we identified several limitations and challenges:
• Computational Demands: Our system required more pow- erful hardware to maintain real-time performance, partic- ularly in AR settings with complex environments.
• Occasional Inconsistencies: In longer gameplay sessions, some participants noticed occasional inconsistencies in NPC personality or knowledge, highlighting the need for improved long-term memory modeling.
• Learning Curve: Some participants, especially those less experienced with AI systems, initially found the open- ended nature of conversations challenging, suggesting a need for better onboarding or tutorials.
• Ethical Considerations: The highly engaging nature of our AI-driven NPCs raised questions about potential overattachment or addiction, warranting further investi- gation into the ethical implications of highly realistic AI companions in games.

4.6 Implications for AR/VR Game Design

Our results have several important implications for the future of

AR/VR game design:

- **Enhanced Storytelling:** The improved NPC interactions enable more dynamic and personalized storytelling, al- lowing for branching narratives that truly adapt to player choices and emotions.
- **Reduced Development Costs:** While our system requires initial setup, it has the potential to significantly reduce the cost and time required for scripting extensive dialogue trees.
- **New Gameplay Mechanics:** The ability of NPCs to un- derstand and respond to complex player inputs opens up possibilities for new types of puzzles, quests, and social gameplay mechanics.
- **Improved Accessibility:** The natural language interface could make AR/VR games more accessible to players who struggle with traditional game controls.

6. Conclusion

This research demonstrates that by leveraging advanced NLP techniques, it is possible to significantly improve the quality of player-NPC interactions in AR/VR games. The inte- gration of context-awareness, emotional intelligence, and real- time adaptation creates more natural, engaging, and immersive conversations, enhancing the overall gaming experience [1-25].

References

1. Jung, S., Lee, B. J., & Han, I. (2011). Gomez, Ł. Kaiser, and I. Polosukhin,"Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.[31] K. Ito et al.,"The lj speech dataset," 2017.[32] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer,"Crowdmos. ADE DE SÃ, 97.
2. Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv*:1810.04805.
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems, 33,* 1877-1901.
4. Rashkin, H. (2018). Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv*:1811.00207.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
6. Hochreiter, S. (1997). Long Short-term Memory. *Neural Computation MIT-Press.*
7. Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv*:1907.11692, 364.
8. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv*:1707.06347.
9. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In European conference on computer vision (pp. 213-229). *Cham: Springer International Publishing.*
10. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.
11. Bao, S., He, H., Wang, F., Wu, H., & Wang, H. (2019). PLATO: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv*:1910.07931.
12. Sutton, R. S. (2018). *Reinforcement learning: An introduction.* A Bradford Book.
13. Zhang, S. (2018). Personalizing dialogue agents: I have a dog, do you have pets too. *arXiv preprint arXiv*:1801.07243.
14. Azuma, R. T. (1997). A Survey of Augmented Reality. Presence: *Teleoperators and Virtual Environments/MIT press.*
15. Slater, M. (2009). Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society B: Biological Sciences, 364*(1535), 3549-3557.
16. I. Goodfellow, Y. Bengio, and A. Courville, *"Deep Learning," MIT Press,* 2016.
17. LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning. nature, 521*(7553), 436-444.
18. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems,* 25.
19. Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv*:1301.3781, 3781.
20. Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv*:1409.0473.
21. Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.
22. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
23. Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, *speech and signal processing* (pp. 6645-6649). Ieee.
24. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature, 529*(7587), 484-489.
25. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature, 518*(7540), 529-533.