Title

Neuro-Symbolic Persona Grounding: Enhancing LLM-based Personas with Structured Knowledge

Problem Statement

LLM-based personas often lack grounding in real-world knowledge and experiences, resulting in shallow or inconsistent personality representations. This limits their ability to engage in deep, contextually appropriate interactions and maintain coherent personalities across diverse scenarios.

Motivation

Current approaches to persona development typically rely on text-based descriptions or fine-tuning on dialogue datasets, which may not capture the rich, structured knowledge underlying human personalities. Integrating symbolic knowledge representations with neural language models can provide a more robust foundation for persona development, allowing for deeper reasoning and more consistent behavior. By combining the strengths of knowledge graphs and LLMs, we can create personas that are both flexible in language generation and grounded in structured knowledge, leading to more realistic and coherent interactions.

Proposed Method

We propose Neuro-Symbolic Persona Grounding (NSPG), a framework that combines symbolic knowledge bases with neural language models to create more richly grounded personas. The method consists of the following components:

- Knowledge Graph Construction: Build a multi-relational knowledge graph representing common human experiences, beliefs, and social norms.
- Graph-to-Text Encoder: Develop a novel encoder that translates subgraphs into natural language descriptions for conditioning the LLM.
- Persona Creation: Select a subgraph representing core personality traits and experiences for each persona.
- Graph Neural Network: Use a GNN to reason over the subgraph, generating embeddings that capture complex relationships between concepts.
- Cross-Attention Mechanism: Utilize the GNN embeddings to guide the LLM's output through a cross-attention mechanism.
- Dynamic Graph Expansion: Implement a module that retrieves relevant knowledge based on dialogue context, allowing the persona to access a broader knowledge base as needed.
- Symbolic Constraint Checking: Introduce a mechanism that verifies the LLM's outputs against the knowledge graph, rejecting inconsistent responses.
- End-to-End Training: Train the entire system on a dataset of persona-grounded dialogues, optimizing for both language modeling and graph-based reasoning objectives.

Step-by-Step Experiment Plan

Step 1: Data Preparation

1 1. Collect a diverse set of persona descriptions and associated dialogues from existing datasets (e.g., PersonaChat, ConvAI2).
2 2. Manually annotate a subset of these personas with structured knowledge graph representations.
3 3. Develop a semi-automated process to extend this annotation to the full dataset.
4 4. Create a new dataset of knowledge-intensive conversations that require deep persona understanding.

Step 2: Knowledge Graph Construction

1 1. Define the schema for the multi-relational knowledge graph, including entity types (e.g., traits, experiences, beliefs) and relation types.
2 2. Implement a graph database (e.g., Neo4j) to store and query the knowledge graph.
3 3. Populate the graph with common knowledge about human experiences and social norms.
4 4. Develop a mechanism to create persona-specific subgraphs from the main knowledge graph.

Step 3: Graph-to-Text Encoder Development

1 1. Design the architecture for the graph-to-text encoder, using a combination of graph neural networks and transformer-based models.
2 2. Implement the encoder using a deep learning framework (e.g., PyTorch).
3 3. Train the encoder on pairs of subgraphs and corresponding natural language descriptions.

Step 4: LLM Integration

1 1. Select a pre-trained LLM (e.g., GPT-3, BERT) as the base model.
2 2. Implement the cross-attention mechanism to incorporate graph embeddings into the LLM's generation process.
3 3. Develop prompts for the LLM that effectively utilize the graph-encoded information.

Step 5: Dynamic Graph Expansion Module

1 1. Implement a retrieval mechanism that identifies relevant subgraphs based on dialogue context.
2 2. Develop an algorithm to dynamically expand the persona's knowledge graph during conversations.
3 3. Integrate this module with the main NSPG framework.

Step 6: Symbolic Constraint Checking

1 1. Develop a set of rules for checking consistency between LLM outputs and the knowledge graph.
2 2. Implement a verification module that applies these rules to generated responses.
3 3. Create a feedback mechanism that rejects or modifies inconsistent responses.

Step 7: End-to-End Training

1 1. Combine all components into a unified NSPG framework.
2 2. Define a multi-objective loss function that balances language modeling and graph-based reasoning.
3 3. Implement a training loop that optimizes all components jointly.
4 4. Train the model on the prepared dataset, using gradient accumulation and mixed-precision training for efficiency.

Step 8: Evaluation

1 1. Evaluate NSPG on existing persona-based dialogue datasets (e.g., PersonaChat, ConvAI2).
2 2. Assess performance on the new knowledge-intensive conversation benchmark.
3 3. Conduct human evaluations to measure perceived personality depth and realism.
4 4. Perform ablation studies to assess the impact of different components (e.g., graph reasoning, constraint checking).

5 5. Compare NSPG against baseline methods, including standard LLM fine-tuning and previous persona modeling approaches.

Step 9: Analysis and Iteration

1 1. Analyze the results to identify strengths and weaknesses of the NSPG approach.
2 2. Investigate failure cases and inconsistencies in the model's outputs.
3 3. Refine the knowledge graph, encoder, and other components based on the analysis.
4 4. Iterate on the training process, adjusting hyperparameters and model architecture as needed.

Test Case Examples

Baseline Prompt Input

You are a 25-year-old software engineer named Alex who loves hiking and playing guitar. You're talking to a new friend at a party. Friend: 'Hey, I heard you're into music. What kind of stuff do you play?'

Baseline Prompt Expected Output

Hi there! Yes, I'm really into music. I love playing the guitar, especially acoustic. I mostly play indie and folk rock, but I also enjoy experimenting with some blues and jazz. Do you play any instruments yourself?

Proposed Prompt Input

You are a 25-year-old software engineer named Alex who loves hiking and playing guitar. Your knowledge graph includes information about your musical preferences, skill level, and experiences. You're talking to a new friend at a party. Friend: 'Hey, I heard you're into music. What kind of stuff do you play?'

Proposed Prompt Expected Output

Hey! Yeah, music is a huge part of my life. I've been playing guitar for about 10 years now, mostly focusing on fingerstyle acoustic. I'm really into indie folk - artists like Bon Iver and Fleet Foxes have been huge influences. Recently, I've been trying to incorporate some of the techniques I've picked up from hiking into my music. There's this one song I wrote inspired by a sunrise I saw on Mount Rainier last summer. It's got this slow build, kind of like watching the light gradually spread across the valley. Do you have any experience with combining your hobbies like that?

explanation

The baseline response provides a generic answer about playing guitar and musical preferences. In contrast, the NSPG-generated response demonstrates a deeper, more nuanced personality. It includes specific details about Alex's musical journey (10 years of experience, fingerstyle technique), concrete influences (Bon Iver, Fleet Foxes), and even connects the musical interest to the hiking hobby mentioned in the persona description. This showcases how the knowledge graph allows for richer, more interconnected responses that feel more authentically human.

Fallback Plan

If the proposed NSPG method doesn't meet our success criteria, we can pivot our research in several directions. First, we could conduct a detailed error analysis to understand where the model is failing. This might involve categorizing different types of inconsistencies or hallucinations and tracing them back to specific components of our architecture. We could then focus on improving the most problematic areas, such as refining the knowledge graph structure or enhancing the graph-to-text encoding process. Another approach would be to investigate hybrid methods that combine our neuro-symbolic approach with more traditional persona modeling techniques. For example, we could explore ways to integrate retrieval-based methods or meta-learning approaches for faster adaptation to new personas. Additionally, we could shift our focus to analyze the interpretability aspects of our model. By examining how information flows from the knowledge graph to the final output, we might gain insights into how LLMs integrate structured knowledge into their responses. This could lead to a valuable analysis paper on the strengths and limitations of neuro-symbolic approaches in persona modeling, even if our specific implementation doesn't outperform baselines on all metrics.

Ranking Score: 6