Language Model for 20 Questions Game

Manasi Bondalapati
Naveen Subhash Udata
Dheeraj Kumar Goli
Vikramadithya Pabba

Northeastern University
Boston, MA 02115

Abstract

This project focuses on developing an AI-based gamebot capable of playing the 20 Questions game using a state-of-the-art language model. The gamebot leverages the Meta Llama 3.1 Instruct model to interact with users by asking relevant questions and making educated guesses based on responses. The objective is to create an intelligent agent that can effectively narrow down the category of a chosen word (place, person, or thing) through a series of yes/no questions and ultimately guess the word correctly. This report outlines the methodology, implementation, results, and evaluation of the gamebot's performance.

1 Introduction

The 20 Questions game is a classic game where one player thinks of a word, and the other player asks up to 20 yes/no questions to guess the word. With advancements in Natural Language Processing (NLP) and machine learning, this project explores the feasibility of developing an AI gamebot that can autonomously play the game. The gamebot uses a pre-trained language model to generate questions and guesses, aiming to provide a challenging and engaging interactive experience for users.

The model used for this task is the **Meta-Llama-3.1-8B-Instruct** model, a cutting-edge NLP model developed for various instruction-following tasks. The Meta-Llama-3.1-8B-Instruct model is part of a family of large language models built upon the transformer architecture. With 8 billion parameters, this model is designed to handle complex natural language understanding and generation tasks, making it an ideal candidate for interactive applications like the 20 Questions game.

2 Methodology

2.1 Dataset and Data Pre-processing

For this 20-question guessing game project, a traditional dataset is not required in the conventional sense of large, labeled data collections. Instead, the project leverages a pre-trained language model, Meta-Llama-3.1-8B-Instruct, which has already been trained on extensive text data. The game is interactive, relying on user inputs—specifically, the answers to the model's questions—rather than processing a static dataset.

Architecture of Meta-Llama-3.1-8B-Instruct

The architecture of Meta-Llama-3.1-8B-Instruct is based on the transformer model, introduced by

Vaswani et al. in 2017. Transformers are the foundation of modern NLP models, allowing for the processing of input data in parallel rather than sequentially, which is a significant advancement over previous recurrent neural network (RNN) architectures.

Key Components of the Architecture:

1. **Multi-Head Self-Attention Mechanism:**
    - The self-attention mechanism allows the model to focus on different parts of the input sequence when making predictions. The "multi-head" aspect refers to the model's ability to capture various relationships in the data simultaneously by using multiple attention heads.
2. **Layer Normalization:**
    - Layer normalization is applied to stabilize the training of the model by normalizing the inputs to each layer, improving convergence during training.
3. **Feed-Forward Neural Networks:**
    - Each attention layer is followed by a position-wise fully connected feed-forward network. This network helps the model learn complex transformations of the data.
4. **Positional Encoding:**
    - Since transformers do not inherently capture the order of input tokens, positional encodings are added to the input embeddings to provide information about the position of tokens in the sequence.
5. **Decoder-Only Model:**
    - Unlike the original transformer architecture, which uses both an encoder and a decoder, the Meta-Llama-3.1-8B-Instruct model employs a decoder-only architecture. This design is optimized for text generation tasks, where the model predicts the next token in a sequence based on the previous tokens.

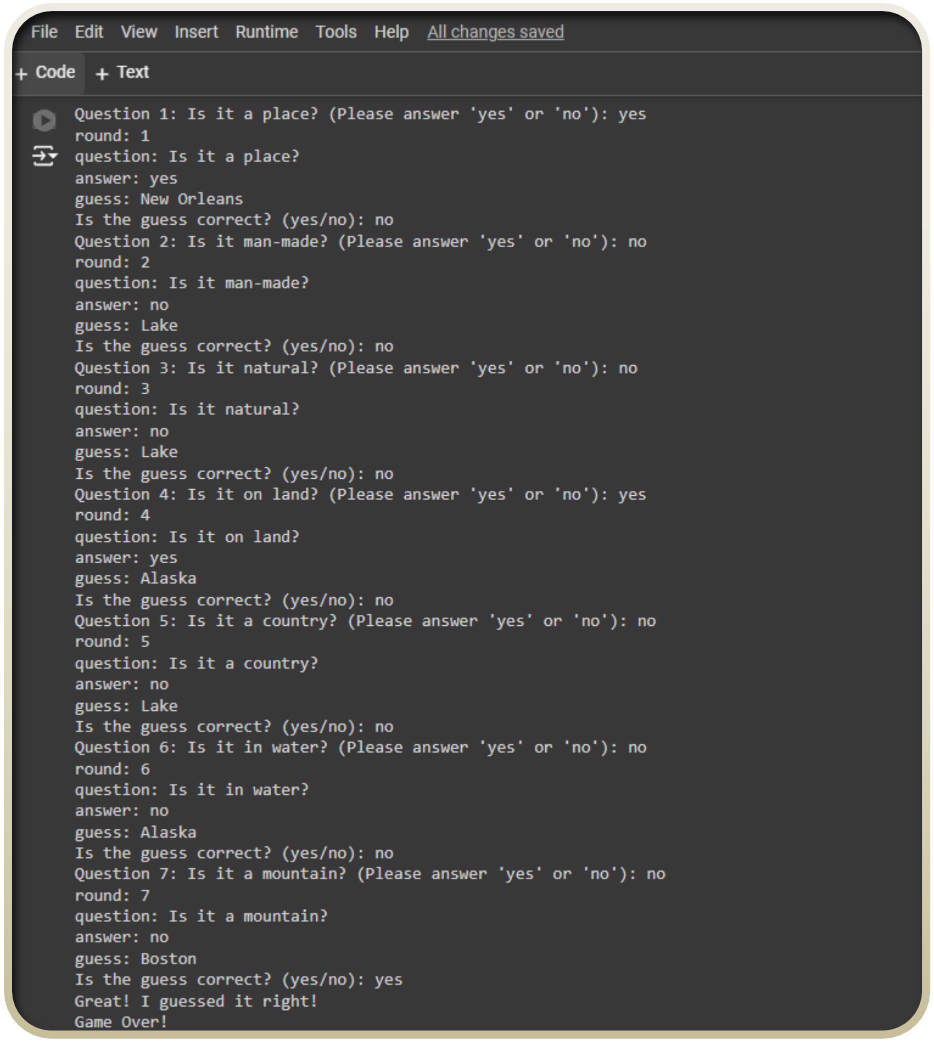
2.2 Implementation

The core of the gamebot's functionality is based on the Meta Llama 3.1 (8B) Instruct model, accessed through the Hugging Face Transformers library. The model was chosen for its advanced language understanding and generation capabilities.

1. **Model Loading**:
    - **Tokenization**: The Auto Tokenizer class converts text into model-compatible input tokens.
    - **Model Initialization**: The AutoModelForCausalLM class loads the pre-trained language model with torch.bfloat16 precision and device mapping to handle different hardware configurations.
2. **Gamebot Design**:
    - **Roles**: The gamebot performs two primary roles:
        - **Questioning**: Asks yes/no questions to narrow down the category of the word.
        - **Guessing**: Makes educated guesses based on previous questions and answers.
    - **Tracking**: Maintains a set of guessed words to avoid repetition and improve guessing accuracy.
3. **Simulation Loop**:
    - **Interaction**: The gamebot interacts with the user in a loop for up to 20 rounds. It alternates between asking questions and making guesses based on user responses.
    - **Termination**: The game ends if the bot correctly guesses the word or if all 20 questions are used.

3 Results

For testing, we took “Boston” as the word for the model to guess. The following are the results we got from the model. The model gave the correct guess within 7 guesses.



```
Question 1: Is it a place? (Please answer 'yes' or 'no'): yes
round: 1
question: Is it a place?
answer: yes
guess: New Orleans
Is the guess correct? (yes/no): no
Question 2: Is it man-made? (Please answer 'yes' or 'no'): no
round: 2
question: Is it man-made?
answer: no
guess: Lake
Is the guess correct? (yes/no): no
Question 3: Is it natural? (Please answer 'yes' or 'no'): no
round: 3
question: Is it natural?
answer: no
guess: Lake
Is the guess correct? (yes/no): no
Question 4: Is it on land? (Please answer 'yes' or 'no'): yes
round: 4
question: Is it on land?
answer: yes
guess: Alaska
Is the guess correct? (yes/no): no
Question 5: Is it a country? (Please answer 'yes' or 'no'): no
round: 5
question: Is it a country?
answer: no
guess: Lake
Is the guess correct? (yes/no): no
Question 6: Is it in water? (Please answer 'yes' or 'no'): no
round: 6
question: Is it in water?
answer: no
guess: Alaska
Is the guess correct? (yes/no): no
Question 7: Is it a mountain? (Please answer 'yes' or 'no'): no
round: 7
question: Is it a mountain?
answer: no
guess: Boston
Is the guess correct? (yes/no): yes
Great! I guessed it right!
Game Over!
```

4 Discussion

The gamebot's performance was generally satisfactory, with a few observations:

- **Effectiveness of Questions**: The model's ability to generate insightful questions was crucial for narrowing down the word category. However, the quality of questions could be further improved with more refined prompt engineering.
- **Guess Accuracy**: The gamebot's guessing capability improved with more context from previous questions and answers. Ensuring diversity in guesses and handling edge cases are areas for future enhancement.
- **User Interaction**: The interaction was intuitive, but additional features such as dynamic question adjustment or adaptive learning could enhance the overall experience.

5 Conclusion

The AI-Based 20 Questions Gamebot project achieved its goal of creating an intelligent agent capable of playing the 20 Questions game. By integrating the Meta Llama 3.1 Instruct model, the gamebot effectively interacted with users, generating relevant questions and making educated guesses. The project highlights the potential of leveraging advanced language models for interactive applications. Future work will focus on refining the model's question generation, enhancing guessing accuracy, and exploring additional features to improve user engagement and game performance.

6 Acknowledgement

This project was completed for CS6120 under the instruction of Professor Uzair Ahmad.

We want to thank the developers of the Hugging Face library for their invaluable tools and resources.

References

[1] Radford et al. (2019), "Language Models are Few-Shot Learners"

[2] Brown et al. (2020), "Language Models as Knowledge Bases?"