

Author: Kerem Kazan

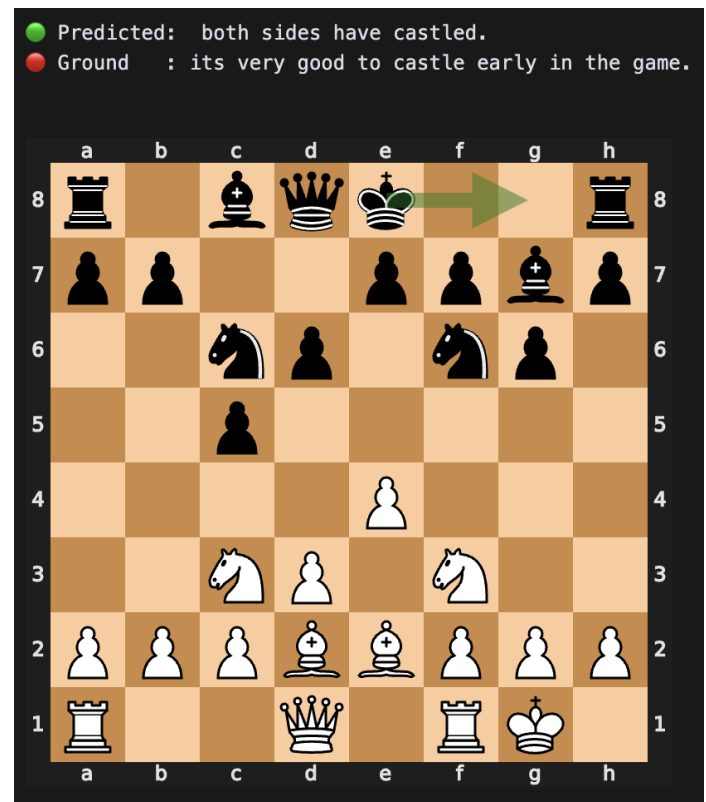
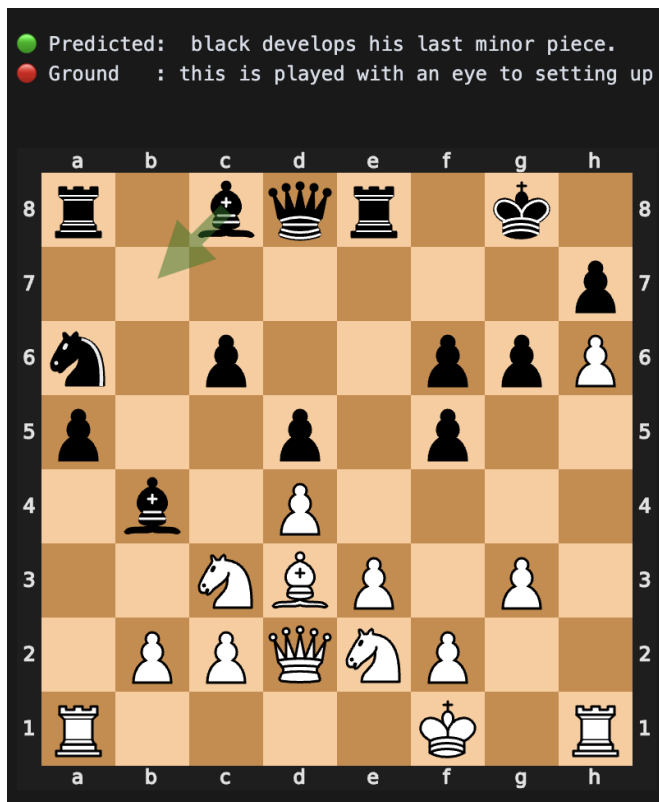
Title: Chess Commentator Transformer

Motivation: Chess board analysis is an endeavor as old as time. There are plenty of analysis engines out there, such as [Stockfish](#). These engines assign numerical scores to boards by simulating moves and attempting to understand which paths lead to certain victory, or certain defeat. These engines will never show you the idea, or the thinking behind these moves - like a teacher would. Some “coach bot”s offer rules-based hand-coded algorithms that can be reduced to a series of nested if-statements that will never capture the deep, interesting patterns that keep emerging in a game as complex as chess. So the issue remains: we need a natural language chess bot that can offer comments on chess boards and moves.

Dataset: Almost all chess datasets match boards and moves to winners or scores, but not comments. Upon inspecting prior art, many references to the [Gameknot.com](#) chess forum games came up, along with outdated code that was meant to scrape them. Solution: write the scraping code from scratch. The result was 12449 games and 278,000 rows of (board,move,comment) tuples. This dataset is not publicly available, so I'm only able to share a small sample of 500 rows: <https://tinyurl.com/yck7twmc>

Approach: Use an encoder-decoder architecture - specifically BERT and GPT2. Manually sequence the (board,move) inputs and train a compact tokenizer. Use a world-level tokenizer of size 1000 for the comments. Train on A100.

Results: The model is coherent, and able to make useful comments. However, it's prone to repeating common phrases from the dataset, such as “I think this is a good move”. Below are 2 cherry-picked examples of good board understanding.



Author: Kerem Kazan

Title: Chess Commentator - Transformer

2 Minute Video: <https://youtu.be/fw4TVnBHX3U>

15 Minute Video: <https://youtu.be/TRrf12bLbKY>