# NLP for Chess

Mateusz Tabaszewski
Bartłomiej Pukacki
Krzysztof Weber
Adam Mielniczuk

# Table of Contents

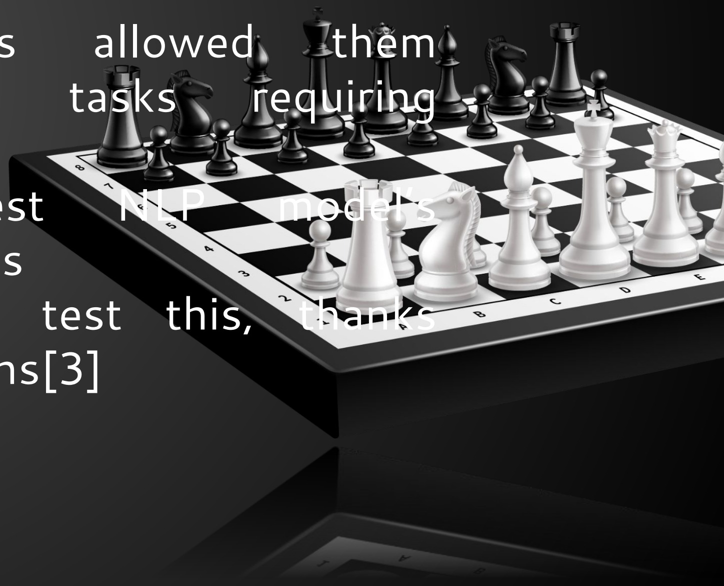


# Division of Responsibilities

- Exploration of Human Chess Games, BERT for Move Legality Classification, Presentation – **Mateusz Tabaszewski**
- NLP for Chess-Playing (GPT-2) – **Bartłomiej Pukacki**
- NLP for Chess-Playing (chessGPT), NLP Models for Opening Recognition – **Krzysztof Weber**
- NLP for Chess-Playing (GPT-2 – Large) – **Adam Mielniczuk**
- Paper/Report – **Collaborative**

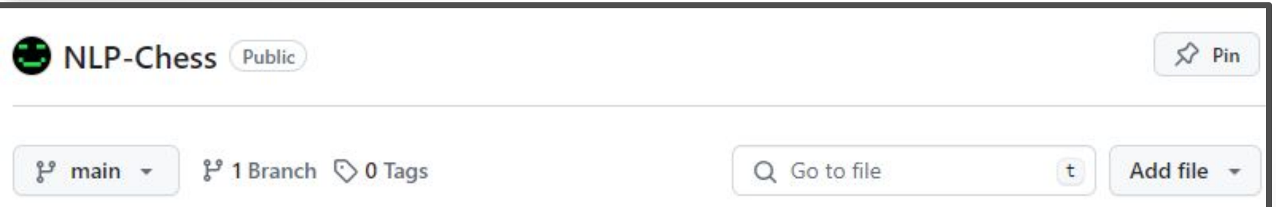
# Introduction

- Humans have used board games to test each other’s logical and strategic abilities for thousand of years – Royal Game of Ur is almost 4000 years old[1]
- Development of NLP models has allowed them to perform very well in **some** tasks requiring logical thinking (code generation)[2]
- It makes sense to try and test NLP model’s logical abilities in the context of board games
- Chess is one of the best games to test this, thanks to its popularity and already existing solutions[3]



# Exploration of Human Games

- Exploration of how humans play chess may allow us to discover good metrics for judging the model's performance
- It might allow us to compare the models with humans
- Dataset of chess games is publicly available on Lichess[4]
- We based our analysis on multiple attributes like: Player's ELO, performed moves, StockFish evaluation and more...
- Details available on the project's repository[5]




# Exploration of Human Games

- Comparison of Human and Random Moves according to the StockFish Engine
- Humans play consistently better than random players
- According to the StockFish Engine most humans play moves which lead to worse evaluations

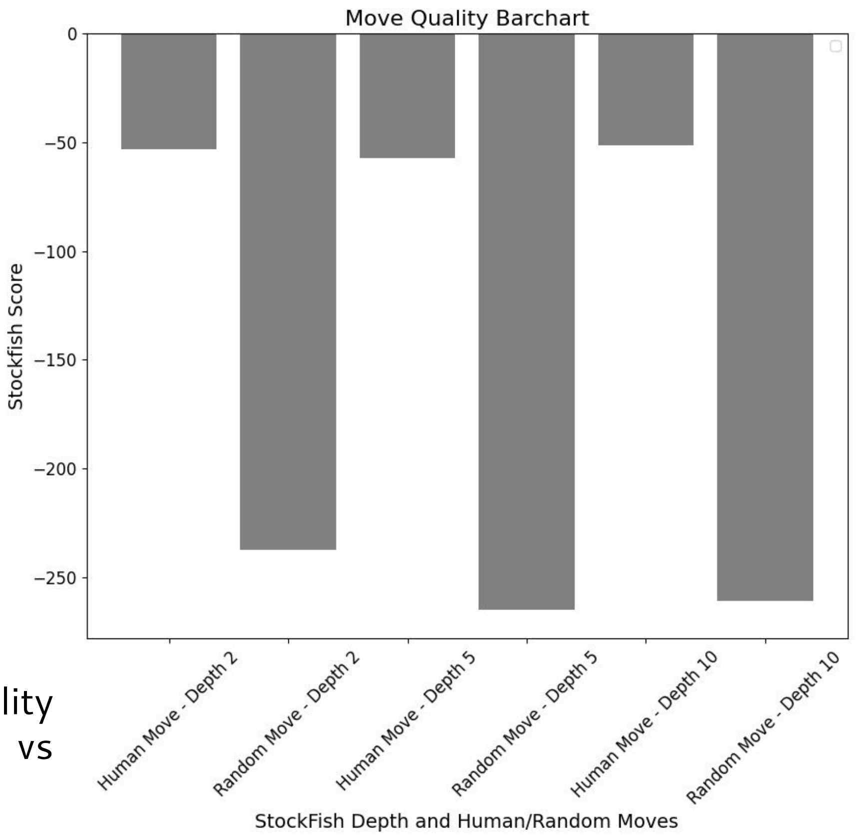



Fig. 1. - Stockfish move quality evaluation plot of human vs random moves

# Exploration of Human Games

- Distributions of Human and random move quality evaluations are distinctly different
- Humans tend to perform very bad moves much more rarely than random players

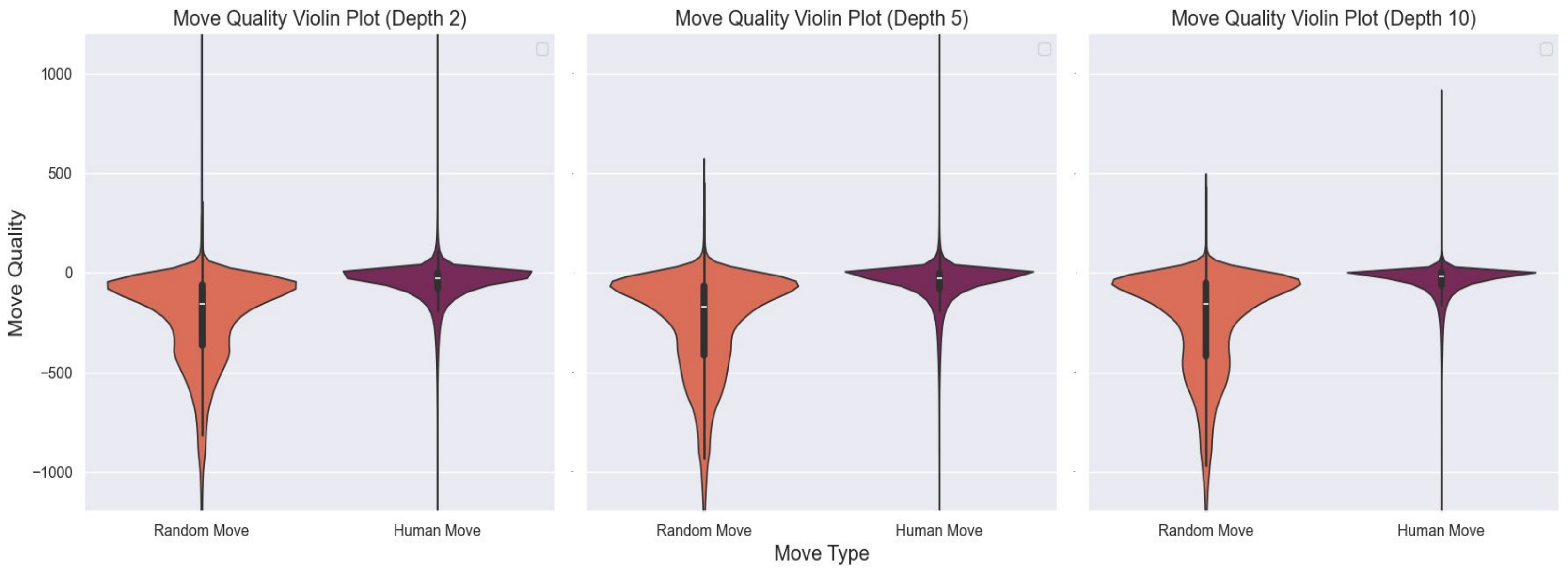



Fig. 2. – Stockfish move quality violin plots for human vs random players

# Exploration of Human Games

- Comparison of most popular moves and their StockFish evaluations depending on the player's experience level
- More experienced players tend to perform more beneficial moves
- With the exception of g7g6 for Novices

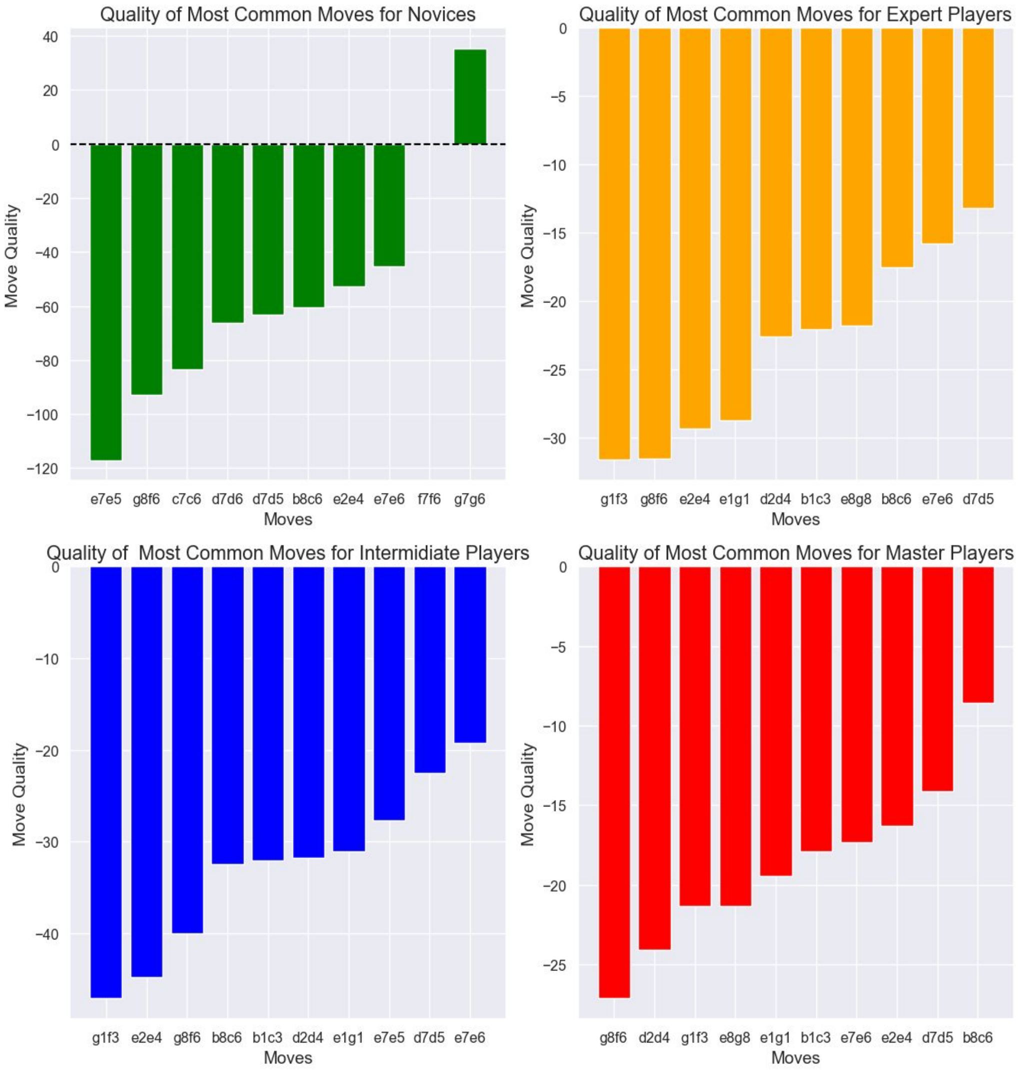
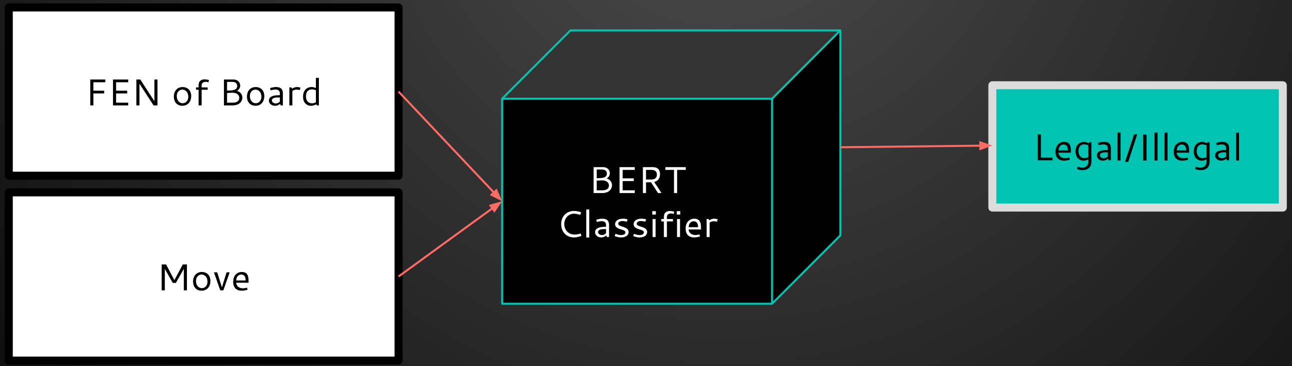



Fig. 3. – Stockfish move quality evaluation plot depending on player's experience for most common moves for each experience level

# BERT for Move Legality Classification

- Understanding rules of chess is crucial for correct and high-level play
- We have decided to test if it's possible to train a BERT model to try to predict legality of presented moves



# BERT for Move Legality Classification

- BertForSequenceClassification from the transformers package was trained for 40 epochs on 10 000 training examples filled with both legal and illegal moves
- The model achieved performance of **85.0%** on the test set
- Clearly, NLP models can learn, at least to some extent, the rules governing a game of chess

```
h 33: Validation loss: 0.6078 Validation accuracy: 87.0%
poch: 34 Training loss: 0.0472 Training accuracy: 98.24%
Epoch 34: Validation loss: 0.5705 Validation accuracy: 87.2%
Epoch: 35 Training loss: 0.0501 Training accuracy: 98.28%
Epoch 35: Validation loss: 0.5501 Validation accuracy: 88.4%
Epoch: 36 Training loss: 0.037 Training accuracy: 98.61%
Epoch 36: Validation loss: 0.5777 Validation accuracy: 88.0%
Epoch: 37 Training loss: 0.0348 Training accuracy: 98.72%
Epoch 37: Validation loss: 0.5808 Validation accuracy: 87.8%
Epoch: 38 Training loss: 0.0429 Training accuracy: 98.52%
Epoch 38: Validation loss: 0.6649 Validation accuracy: 86.4%
Epoch: 39 Training loss: 0.0377 Training accuracy: 98.67%
Epoch 39: Validation loss: 0.5474 Validation accuracy: 88.6%
```

```
torch.save(model, f"{main_directory}/{model_path}")
```

## Evaluation

```
model = torch.load(f"{main_directory}/{model_path}")
```

```
evaluate("Test", model, test_loader, criterion, num_test_examples,
```

```
Epoch Test: Test loss: 0.7671 Test accuracy: 85.0%
```

# NLP for Chess-Playing

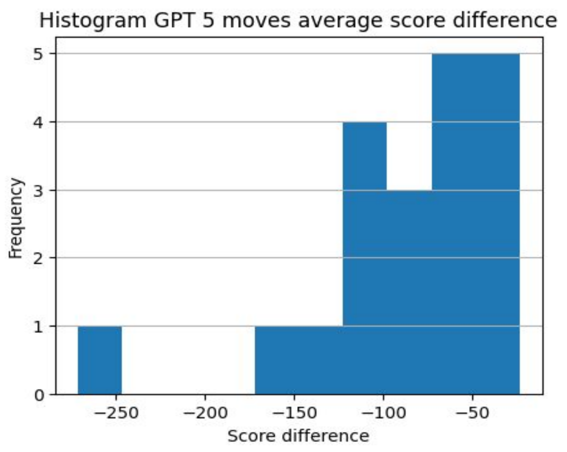
- Goal: test the ability of some generic/specialised language models in predicting the next move in SAN notation.
- Check legality and quality of the output.
- Compare to Random/Human player.
- Try to enforce legal moves using the *force_word_ids* argument.

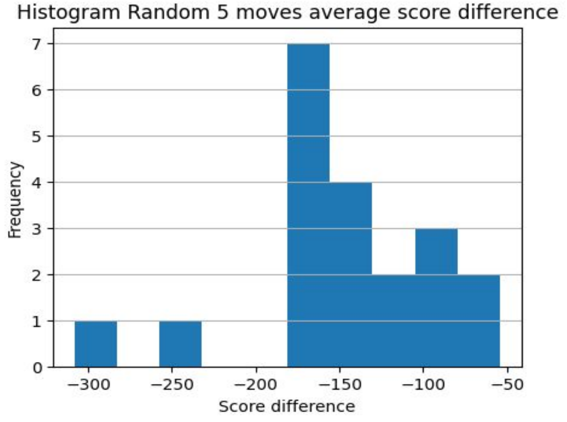
```
query = f"Provide the next move in the chess game. Only provide the move, no move numbers. {moves}"
```

```
['1. e4 e5 2. d4 d5 3. exd5 exd4 4. Qxd4 c5 5. Qe4+ Ne7 6. Bg5 f6 7. Nf3 fxg5 8.
 '1. e4 c5 2. Nf3 e6 3. d4 d5 4. exd5 exd5 5. Ne5 a6 6. Qh5 Nf6 7. Qxf7# 1-0',
 '1. d4 d5 2. Nc3 Bd7 3. e4 dxe4 4. Nxe4 Nc6 5. c3 a6 6. Qf3 g6 7. Bd3 Bg7 8. Ne
 '1. e4 Nc6 2. d3 Nd4 3. Nd2 Ne6 4. Ngf3 Nf4 5. g3 Ng6 6. Bg2 Nf6 7. O-O d6 8. R
 '1. e4 Nh6 2. Nf3 e6 3. d4 d5 4. e5 Nc6 5. Bxh6 gxh6 6. c3 Rhg8 7. g3 f6 8. Bb5
```

# NLP for Chess-Playing (GPT-2)

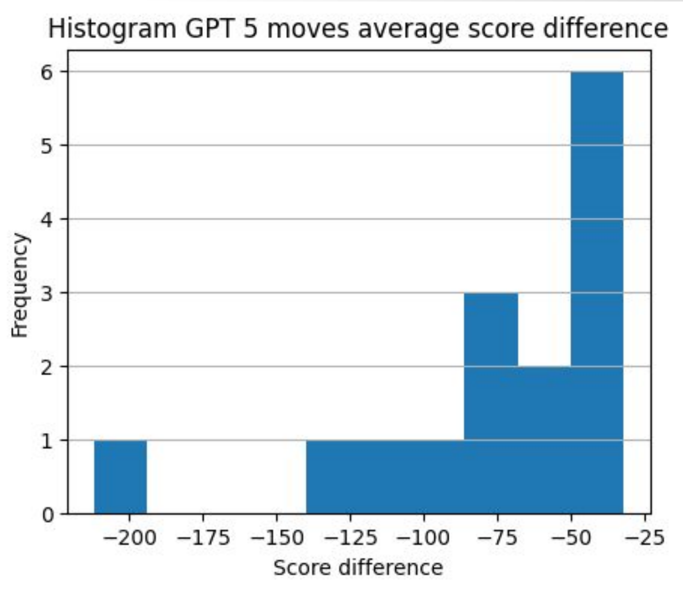
- Consistently better than Random.
- Up to 14–30% legal moves made (depends on previous sequence length) without fine tuning.
- Forcing legal moves does not help the model select valid moves due to inappropriate default tokenization.
- Much worse than a human player.



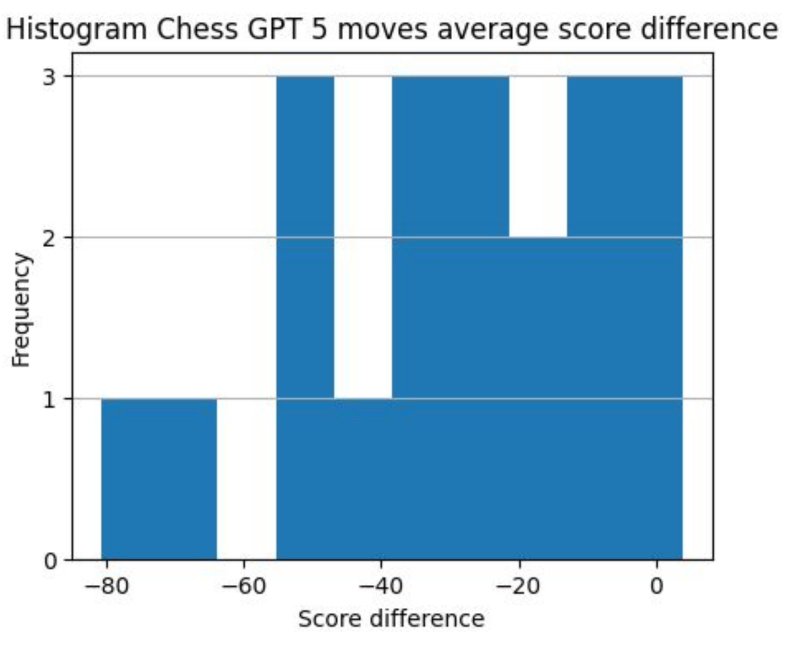
# NLP for Chess-Playing (GPT-2 - Large)

- Consistently performs better player
- Minimally better Stoc than GPT-2
- Noticeably better at p moves than GPT-2
- Still below human-level abilities



# NLP for Chess-Playing (chessGPT)

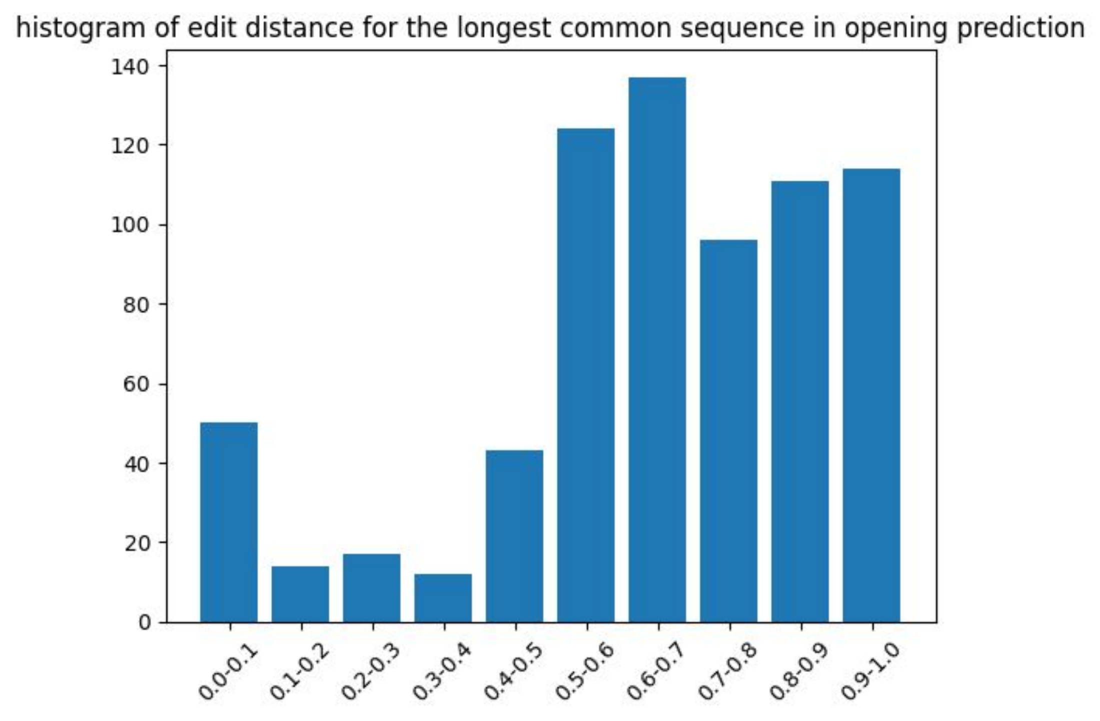
- Greatly outperforms GPT-2 - Large
- Often matches or human-level gameplay
- For short se performs illegal moves




| | Player | Average score difference | % legal moves | % best moves | % above average moves | % worst moves |
|---|---|---|---|---|---|---|
| Aggregate results | GPT-2 | -214.54 | 22.77% | 6.76% | 72.78% | 1.10% |
| | GPT-2 - Large | -213.43 | 40.87% | 9.74% | 77.72% | 0.74% |
| | chessGPT | -69.92 | 98.82% | 31.69% | 95.41% | 0.13% |

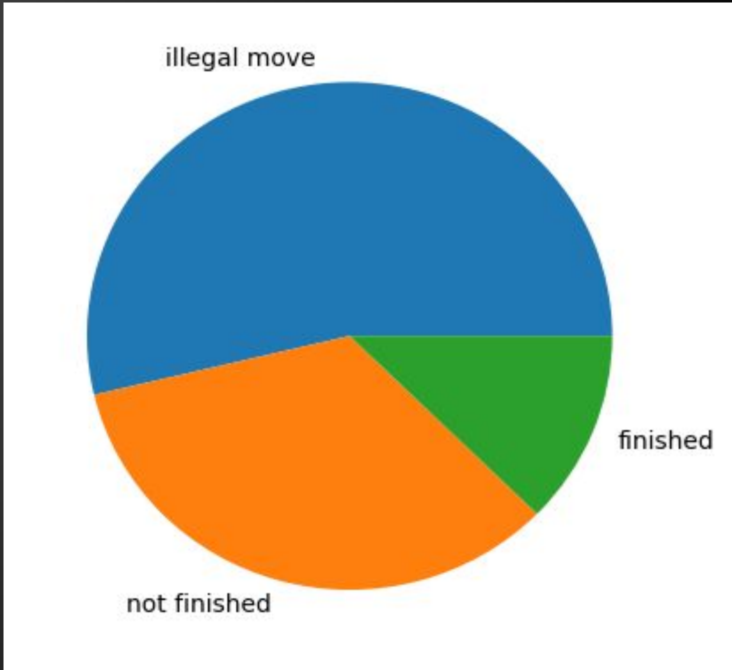
# NLP Models For Opening Recognition

- chessGPT can be asked to recognise chess openings based on a few starting moves
- The model performed well for well-known openings, but made errors for rarely-played openings



# NLP Models For Opening Recognition

- chessGPT also makes mistakes when asked to generate a whole sequence of moves as a game
- Oftentimes thinks the game is finished when it is not



# Conclusions

- Language Models can be used for chess-playing
- More general models outperform random players but achieve significantly below human-level performance
- More specialized chess models (chessGPT) can perform at a human-level but often make mistakes when asked to generate a longer sequence of moves
- Future research could explore new state-of-the-art LLMs and compare them with older, simpler solutions like GPT-2 or try to mitigate errors when generating longer sequences

# Sources


*[1] https://en.wikipedia.org/wiki/Royal_Game_of_Ur*
*[2]Huang, D., Bu, Q., Zhang, J. M., Luck, M., & Cui, H. (2023). AgentCoder: Multi-Agent-based Code Generation with Iterative Testing and Optimisation. ArXiv. /abs/2312.13010*
*[3] Feng, X., Luo, Y., Wang, Z., Tang, H., Yang, M., Shao, K., Mguni, D., Du, Y., & Wang, J. (2023). ChessGPT: Bridging Policy Learning and Language Modeling. ArXiv. /abs/2306.09200*
*[4] https://database.lichess.org*
*[5] https://github.com/MatTheTab/NLP-Chess/tree/main*
*[6] GPT-2 - Large on Huggingface: https://huggingface.co/openai-community/gpt2-large*
*[7] GPT-2 on Huggingface: https://huggingface.co/openai-community/gpt2*
*[8] chessGPT on Huggingface: https://huggingface.co/Waterhorse/chessgpt-base-v1*
*[9] DeLeo, M., & Guven, E. (2022). Learning Chess With Language Models and Transformers. ArXiv. https://doi.org/10.5121/csit.2022.121515*
*[10] Chess openings website: chessopenings.com*
*[11] NLP for Chess Paper: https://github.com/MatTheTab/NLP-Chess/blob/main/results/NLP%20in%20Chess_%20A%20Comprehensive%20Exploration%20of%20the%20Abilities%20of%20Language%20Models%20in%20Game-Playing.pdf*


THANK YOU
For Your Attention