Meet dataChess and CARLSy: Towards the Explanation of Chess Plays

Anonymous ACL submission

Abstract

We introduce dataChess, a curated dataset of annotated chess games, and CARLSy, a model designed to explain the quality of chess moves through natural language. By fine-tuning a Large Language Model, we developed a specialized model for chess commentary generation that leverages recent advancements in Natural Language Processing. Our evaluation—both automatic and through a human study—demonstrates that our model produces commentary on par with state-of-the-art systems. However, it also reveals key challenges that can be addressed to enhance quality and reduce variability in the generated explanations.

1 Introduction

The game of chess has long been a subject of fascination for enthusiasts and a topic of study for researchers.

Chess provides a structured, data-rich environment with symbolic representation, making it an ideal testing ground for evaluating the capabilities of Large Language Models (LLMs). In this context, AI-generated chess commentary emerges as a promising approach to improving the accessibility and interpretability of chess analysis.

In this paper, we introduce dataChess, a curated dataset of annotated chess games. We build dataChess upon a portion of the dataset introduced by Feng et al. (2023) by converting Portable Game Notation (PGN) files into CSV format while enhancing the data with additional features, including attack annotations, piece positioning, tercile-based comment length classification, and comprehensive data cleansing, elevating the dataset's quality and usability. We also explore different approaches to leveraging LLMs for explaining the quality of chess moves. This leads to the development of the Chess Annotation and Recommendation system (CARLSy), a system that provides natural language explanations of chess moves and their strategic quality. We evaluate the performance of CARLSy using automatic metrics, along with a human evaluation study. Our results indicate that CARLSy performs on par with state-of-the-art conditioned models while also highlighting key challenges, such as maintaining consistency in commentaries, and improving contextual awareness in move explanations, that require further exploration. All scripts, code and data are publicly available[1], which is compatible with the original access conditions of the dataset of Feng et al. (2023).

2 Related Work

In chess commentary generation, Jhamtani et al. (2018) introduced an end-to-end neural model that leveraged move, score, and threat characteristics to create annotations. Although their approach produced commentary comparable to human annotations, it struggled with predicting future developments. Zang et al. (2019) addressed this limitation by integrating a neural chess engine trained through supervised learning and self-play. This allowed the system to analyze alternative moves and anticipate future positions, significantly improving the quality, context, and planning aspects of its commentary.

Lee et al. (2022) took a different approach by combining symbolic reasoning with language models. They developed a tag extraction model that identified key chess concepts, which were then used to control text generation. While their system showed improved coherence and was preferred over earlier models, it still suffered from logical inconsistencies.

Considering chess dataset, Feng et al. (2023) introduced the previously mentioned dataset consisting of game records, annotated commentaries, and chess-related conversations, which is at the basis of dataChess.

[1] https://anonymous.4open.science/r/CARLSy-567D

180484 | 1. e4 e6 2. d4 d5 3. e5 c5 4. c3 Nc6 5. Nf3 Qb6 | 6. Bd3 | White R_a1 N_b1 B_c1 Q_d1 K_e1 R_h1 P_a2 P_b2 P_f2 P_g2 P_h2 P_c3 B_d3 N_f3 P_d4 P_e5 Black p_c5 p_d5 q_b6 n_c6 p_e6 p_a7 p_b7 p_f7 p_g7 p_h7 r_a8 b_c8 k_e8 b_f8 n_g8 r_h8 |White B_d3$p_h7 P_d4$p_c5 Black p_c5$P_d4 q_b6$P_b2 n_c6$P_d4 n_c6$P_e5 | black wouldbe mistaken to try and win the d4 pawn. after 6 ... cxd4 7 cxd4 nxd4 8nxd4 qxd4?? 9 bb5 ! this uncovers an attack on black's queen and check'sthe black king. | [LARGE]

180485 | 1. e4 e6 2. d4 d5 3. e5 c5 4. c3 Nc6 5. Nf3 Qb6 6. Bd3 | 6... Bd7 | White R_a1 N_b1 B_c1 Q_d1 K_e1 R_h1 P_a2 P_b2 P_f2 P_g2 P_h2 P_c3 B_d3 N_f3 P_d4 P_e5 Black p_c5 p_d5 q_b6 n_c6 p_e6 p_a7 p_b7 b_d7 p_f7 p_g7 p_h7 r_a8 k_e8 b_f8 n_g8 r_h8 |White B_d3$p_h7 P_d4$p_c5 Black p_c5$P_d4 q_b6$P_b2 n_c6$P_d4 n_c6$P_e5 | preparing for cxd4 | [SMALL]

180487 | 1. e4 e6 2. d4 d5 3. e5 c5 4. c3 Nc6 5. Nf3 Qb6 6. Bd3 Bd7 7. dxc5 Bxc5 8. O-O f6 9. b4 Be7 10. b5 Nxe5 11. Nxe5 fxe5 12. Qh5+ | 12... Kd8 | White R_a1 N_b1 B_c1 R_f1 K_g1 P_a2 P_f2 P_g2 P_h2 P_c3 B_d3 P_b5 Q_h5 Black p_d5 p_e5 q_b6 p_e6 p_a7 p_b7 b_d7 b_e7 p_g7 p_h7 r_a8 k_d8 n_g8 r_h8 | White B_d3$p_h7 Q_h5$p_e5 Q_h5$p_h7 Black q_b6$P_f2 q_b6$P_b5 b_d7$P_b5 | g6 is not an option because bxg6 and black cannot recapture without losing a rook, so white gains a pawn and board space for free | [MEDIUM]

Table 1: Examples of three data points in dataChess

3 Building dataChess

In the development of our initial models, we opted to leverage a portion of the dataset introduced by Feng et al. (2023). Their dataset encapsulates a wealth of chess game data, including **moves**, **positions**, and associated **commentary**. We developed a parser capable of extracting information from their PGN files, by iterating through each file, extracting chess positions in Algebraic Notation[2] along with their corresponding commentary.

After the first experiments, it became evident that the created models lacked an understanding of the chess positions given to it, often leading to the generation of commentary that did not make much sense. To further enhance the model's understanding of each chess position, we drew inspiration from the work of (Lee et al., 2022), and expanded our dataset by incorporating the current square of each piece and the attacks (a legal move that captures an opponents' piece), present in the position, providing the model with richer context during both the fine-tuning and inference stages. We also separated the last move made from the rest of the game.

Then, we began our data cleansing by removing entries with very small commentary (comments with less than 16 characters), as they lacked sufficient information for the model to learn from. Next, we used the Python library langdetect[3] to filter out all entries with non-English comments. We also removed any empty comments to eliminate blank entries that could skew the results and converted all comments to lowercase. We also divided all the comments into tertiles according to their size and labeled them as "Small" (fewer than 54 characters), "Medium" (between 54 and 116 characters) and "Large" (longer than 116 characters). We refer to this dataset as "dataChess" and we can find an excerpt of it in Table 1.

dataChess contains 302994 data points and the number of characters present in the Algebraic Notation, commentaries, moves, positions and attacks can be found in Table 2.

One major problem we had during this project was the lack of high-quality data for constructing our dataset. To address this, we conducted a small experiment using ChatGPT to augment our dataset. We provided it with ten high-quality

[2]Algebraic Notation is the most widely used method of recording chess moves. It uses letters and numbers to represent each square on the chessboard, with ranks (rows) numbered 1 to 8 and files (columns) labeled *A* to *H*. Pieces are represented by their initial letters except the pawns which do not have any letter associated with them, and moves are indicated by the destination square. For example, Ke4 means the king moved to the e4 square.

[3]https://pypi.org/project/langdetect/

Columns	Min	Max	Mean
Algebraic Notation	0	7447	224.597
Commentary	16	3024	122.270
Move	5	13	7.806
Positions	22	172	134.544
Attacks	12	372	57.543

Table 2: Final Dataset Characterization (Min, Max and Mean are Min/Max/Mean size in number of characters)

Dataset	BLEU	ROUGE	METEOR
dataChess	0.8588	0.13851	0.09051
Medium	0.8616	0.14449	0.09854
Medium	**1.2483**	**0.14506**	**0.10306**

Table 3: Best models' results

game/commentary pairs from our dataset as examples and asked it to generate commentary for new positions. Although this experiment had promising results, ChatGPT still made logical mistakes in some of its commentary. This meant that to use these generated comments we would have to manually review and filter them to decide which ones were suitable to introduce to the dataset. Therefore, we decided not to use them.

4 CARLSy

We chose the FLAN-T5 (Chung et al., 2022) model, along with its corresponding tokenizer. The decision was influenced by the model being extremely efficient resource-wise, competing with much larger models parameter-wise in the MMLU Benchmark (Hendrycks et al., 2021).

We made several experiments. We used Flan-T5 Small and Flan-T5 Base, and tested them with different hyperparameters. Using larger versions of this model and a more extensive hyperparameter tuning process could improve its ability to analyze and generate commentary on chess positions, but we lacked the computational resources to test this. We also finetuned the tokenizer as it soon became evident that the tokens generated by the FLAN-T5 tokenizer were unable to capture the chess moves and positions in a meaningful manner. This occurred because the original tokenizer splits each chess move into multiple tokens, hindering the performance of the fine-tuned model by failing to effectively tokenize the input data.

Considering the usual metrics: BLEU (Papineni et al., 2002) (in this case SacreBLEU), METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004), best results (with Flan-T5 Base and the tokenizer finetuned on the first version of dataChess) can be found in Table 3. We used a test set with 1K data points; the remainder of the dataset was split 90% train / 10% validation. Interestingly, we found that "Medium"-sized comments generally have higher quality than shorter and longer ones, as they strike a balance between brevity and depth. Shorter comments often lack sufficient detail or context, making them less informative or impactful while longer comments usually become overly detailed, losing focus of the key aspects of the position. As a result, we configured our final model to only generate medium-sized comments.

The hyperparameters used to train the best model (CARLSy), can be seen in Table 4.

Parameters	Value
Model	FlanT5-Base
Batch Size	8
Gradient Accumulation Steps	4
Learning Rate	$3e^{-4}$
Max Generation Length	200
Number Epochs	8
Weight Decay	0.01

Table 4: CARLSy hyperparameters

5 Human Evaluation

We conducted a human evaluation study through a Google Form, with two main tasks, described next.

5.1 Task 1: Assessing Chess Knowledge

The first task consisted of three multiple-choice questions. In each question, participants were shown a chess position with a checkmate opportunity and asked to select the best move from the three options provided in algebraic chess notation. The purpose of this task was twofold: test participants' chess proficiency and if they could comprehend the Algebraic Chess Notation.

5.2 Task 2: Evaluating Commentary Quality

The second task focused on the quality of the model's commentaries. Participants were presented with ten questions, where they were shown a chess position and five different commentaries for that position, one generated by our model and the other four generated by the three conditioned models

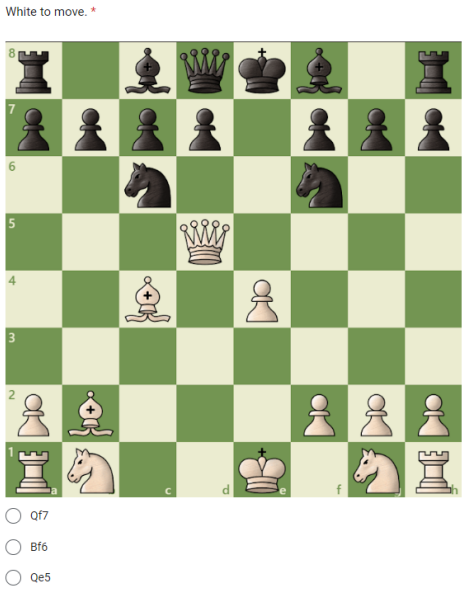

Figure 1: Example question from Task 1

(from now on, Move Description, Move Quality and Comparative) and the unconditioned model (from now on Unconditioned) developed in the work of (Lee et al., 2022). The positions used for this task were the ones present in their paper as we did not replicate their models ourselves.

Figure 2: Example question from Task 2

Participants were then asked to order the comments from the 5 models from best to worst. Additionally, we included an optional section after each position, where they could explain the reasoning behind their choices.

5.3 Results and Discussion

We distributed our questionnaire across various platforms, including chess subreddits[4], Chess.com[5] and Lichess.com forums[6], and Facebook groups.

We received responses from 21 participants out of which three were disqualified from the main task for not passing the initial task. Most of the participants fell within the age range of 18 to 34 years. Among these respondents, the majority classified their chess proficiency as either beginners or intermediate players. Table 5 presents the percentage of participants who prefer our model's commentaries.

Compared Model	Prefer Ours
Move Description	56%
Move Quality	72%
Comparative	44%
Unconditioned	78%

Table 5: Human Evaluation Results

Results demonstrate that our model is on par or better than the compared models from (Lee et al., 2022). Human judges prefer our model over the unconditioned baseline 78% of the time. However, they noted that the models consistently make logical errors, which severely impact their usefulness. In fact, another interesting takeaway is the polarized ranking of our model. Participants frequently ranked it either first or last, indicating a stark contrast in its performance. This pattern suggests that when our model avoids logical errors, it is capable of producing valuable and insightful commentary. However, when it makes logical errors, the quality of its commentary drops dramatically, resulting in extremely poor output.

6 Conclusions

We introduced dataChess and CARLSy, that has the ability to produce highly accurate and insightful comments, but it can not do it consistently enough for it to be reliable. We believe that integrating a chess engine into the generation process could be a valuable step toward reducing the number of errors and the variability of the generated commentary.

[4] https://www.reddit.com/r/chess/, https://www.reddit.com/r/ComputerChess/, https://www.reddit.com/r/chessindia/, https://www.reddit.com/r/Chesscom/

[5] https://www.chess.com/forum

[6] https://lichess.org/forum

7 Limitations

As it currently stands, the available datasets lack sufficient data for a model to be able to learn how to generate meaningful commentary for a game as complex as chess while also containing a lot of data that is not useful for learning. We decided not to use ChatGPT to augment our data, due to the possibility of flooding the dataset with personal bias, but we believe this can be a viable possibility for future projects if done under the right conditions. In the future, using few-shot learning with ChatGPT or other LLM might become a viable option.

Another area for improvement is the use of larger models. Bigger models have the potential to better understand the game of chess and generate more complex and nuanced annotations. However, the increased computational requirements of larger models pose a significant challenge. Addressing this will require access to more powerful hardware.

Additionally, extensive hyperparameter tuning using distributed systems could lead to further performance improvements. Hyperparameter tuning is a critical aspect of optimizing machine learning models and by distributing the computational load across multiple systems we can conduct more extensive searches across the hyperparameter space, increasing the likelihood of finding an optimal configuration.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2023. Chessgpt: Bridging policy learning and language modeling. *Preprint*, arXiv:2306.09200.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1671, Melbourne, Australia. Association for Computational Linguistics.

Andrew Lee, David Wu, Emily Dinan, and Mike Lewis. 2022. Improving chess commentaries by combining language models with symbolic reasoning engines. *ArXiv*, abs/2212.08195.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hongyu Zang, Zhiwei Yu, and Xiaojun Wan. 2019. Automated chess commentator powered by neural chess engine. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5952–5961, Florence, Italy. Association for Computational Linguistics.