Project Proposal: Leveraging Generative Models for Real-Time Chess Commentary

Aarya Devnani
adevnani@usc.edu

Arjun Balamwar
balamwar@usc.edu

Kavish Shah
kavishch@usc.edu

Nimit Jhunjhunwala
njhunjhu@usc.edu

Prisha Panchmia
prishani@usc.edu

Abstract

This project aims to develop a chess commentary system using generative models, incorporating a knowledge graph, retrieval-augmented generation (RAG) pipeline, and fine-tuned language models. The system will analyze chess moves, retrieve relevant historical and strategic data, and generate insightful, context-aware commentary. The system's performance will be evaluated using BLEU scores and human assessments, compared to rule-based engines.

Input	Expected Output
1. e4 d5 2. Nf3 Nc6 3. exd5 Qxd5	"White opens the game in a classical style, controlling the center. Standard continuation is knight f3 or d4 to combat the French Defense".

Table 1: Example Input vs Output for proposed system

3 Introduction

The project, Leveraging Generative Models for Real-Time Chess Commentary, aims to create an intelligent system capable of providing insightful and engaging real-time commentary for chess games.

By leveraging advancements in natural language processing (NLP), specifically large language models (LLMs), our solution enhances the experience of chess enthusiasts across various skill levels. This system integrates a retrieval-augmented generation (RAG) pipeline, fine-tuned models, and a knowledge graph to deliver accurate, context-aware, and player-specific commentary.

Over the course of the project, we collected and curated a comprehensive dataset comprising annotated chess games and commentary sourced from GameKnot's online website. Extensive experimentation with generative models and evaluation metrics demonstrates the system's ability to generate high-quality analyses that adapt to varying game states and player profiles.

This report presents the project's goals, the methodology employed, the results achieved, and a thorough analysis of the system's performance, concluding with insights and potential directions for future work.

4 Related Work

Natural Language Generation (NLG) has been explored across a variety of domains, from rule-based systems to modern learning-based approaches (Reiter et al., 2005)(Reiter et al., 2003). These works have demonstrated practical successes in generating structured text such as weather forecasts (Reiter et al., 2005) and recipe instructions (Yang et al., 2016)(Kiddon et al., 2016). However, generating commentary for chess games poses unique challenges. Unlike structured inputs such as biographies or static records (Lebret et al., 2016), chess game states are dynamic, evolving descriptions that change with every move, requiring real-time understanding and analysis.

A key difference between our work and prior NLG systems lies in the granularity of commentary. Previous efforts like (Wiseman et al., 2017) focus on generating game summaries or aggregate statistics rather than move-by-move analysis. Our project directly addresses this gap by generating insightful commentary for individual moves, as well as multi-move sequences, which demand a deeper pragmatic understanding of the game's progression and strategy.

In addition, earlier efforts such as ROBOCUP match commentary (Chen and Mooney, 2008)(Mei et al., 2015) provide limited datasets (1K events), restricting their scalability. By contrast, our project incorporates a significantly larger dataset collected from GameKnot, enabling the training of robust generative models. Furthermore, while works like (Silver et al., 2016) focus on reinforcement learning for game-playing agents, our approach leverages this game knowledge to enhance commentary generation rather than gameplay performance.

Another unique aspect of our project is the use of advanced methods like retrieval-augmented generation (RAG) pipelines and fine-tuned large language models, which integrate semantic and pragmatic information to improve commentary quality. Unlike earlier datasets such as SUMTIME-METEO (Reiter et al., 2005) and WEATHERGOV (Liang et al., 2009) which faced criticisms for relying on template-based inputs—our work ensures diversity and context-awareness in the commentary.

In summary, while prior NLG systems have demonstrated success in structured text generation, our work differentiates itself by tackling the dynamic, real-time nature of chess commentary. We address the content selection challenges highlighted in (Wiseman et al., 2017) and extend prior efforts by incorporating player-specific profiling, multi-move analysis, and scalable datasets to provide insightful, adaptive commentary.

5 Hypothesis and Evaluation

5.1 Hypothesis

A neural network-based chess engine that incorporates contextual information from the game can generate high-quality, human-like chess commentary.

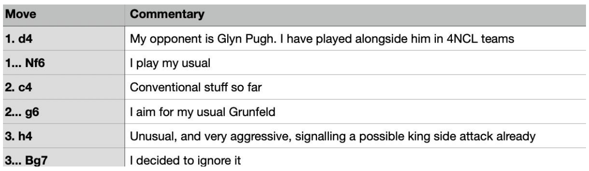
5.2 Evaluation Framework

Measure the quality of generated commentary using metrics like BLEU scores (for text generation quality). The BLEU metric assesses the overlap between generated and reference texts by calculating modified precision across a corpus. In accordance with past studies such as (Zang et al., 2019), and (Jhamtani et al., 2018), we have opted not to use BLEU-4 for evaluation. This decision stems from the fact that BLEU-4 can impose a strict brevity penalty, leading to imbalanced scores, especially when dealing with short outputs that vary in expression. Instead, we will use BLEU-2 to achieve more consistent and reliable results.

6 Approach

6.1 Data

We collected the data from an online website for chess players that provides insights into games as well as a discussion forum, gameknot.com [1]. GameKnot serves as a forum where users can discuss games, strategies, and chess-related topics in a more informal setting. Since GameKnot operates as an open forum, the language and commentary style can be quite informal and vary significantly in grammar and expression. This diverse language use presents challenges for data-driven models that rely on consistency in text patterns. GameKnot provides multiple games which have been annotated with move-by-move commentary. Some of these games have been annotated by the players themselves or by professionals of the game.
We then scraped the HTML files for games, some of which spanned multiple pages. These HTML files were then parsed and combined to form a csv file of each game. The csv files contain rows of moves or move sequences and their corresponding commentary. The total number of csv files generated is 12,427. This dataset was split into training, validation, and test sets using a 7:1:2 ratio with respect to the number of games.



Move	Commentary
1. d4	My opponent is Glyn Pugh. I have played alongside him in 4NCL teams
1... Nf6	I play my usual
2. c4	Conventional stuff so far
2... g6	I aim for my usual Grunfeld
3. h4	Unusual, and very aggressive, signalling a possible king side attack already
3... Bg7	I decided to ignore it

Figure 1: Examples of data stored in the csv scraped from the website

6.2 Methodology

There are three phases i.e. Phase A: Position Analysis, Phase B: Temporal Knowledge Graph Construction and Querying, Phase C: Contextual Language Generation; in our proposed method which would be described in detail below. This multi-phase approach integrates real-time positional analysis, historical game state relationships, and dynamic interactions to create a comprehensive framework for understanding and articulating the complexities of chess gameplay.
The flowchart below gives an overview of the overall methodology:

[1] https://gameknot.com/chess-games-database.pl

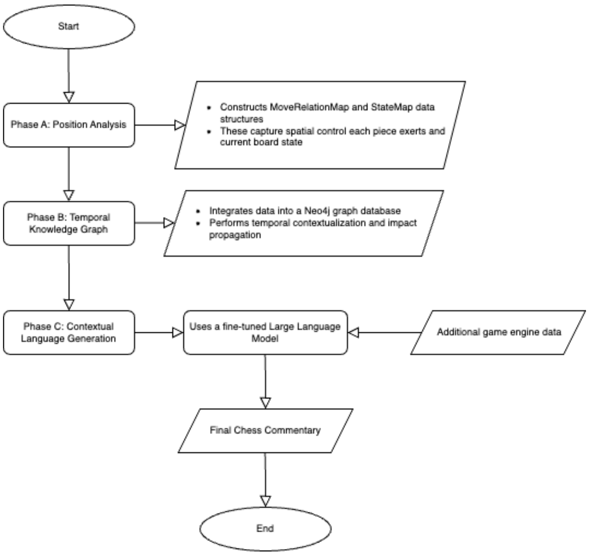

Start
Phase A: Position Analysis
Constructs MoveRelationMap and StateMap data structures
These capture spatial control each piece exerts and current board state
Phase B: Temporal Knowledge Graph
Integrates data into a Neo4j graph database
Performs temporal contextualization and impact propagation
Phase C: Contextual Language Generation
Uses a fine-tuned Large Language Model
Additional game engine data
Final Chess Commentary
End

Figure 2: Flowchart of Methodology

6.2.1 Phase A: Position Analysis

The first phase involves constructing two core data structures: `MoveRelationMap` and `StateMap`.

- `MoveRelationMap` stores positional influence data, where keys represent all chessboard positions and values list all pieces capable of attacking (opponent) or protecting (friendly) that position. This allows us to capture and quantify the spatial control each piece exerts over the board.
- `StateMap` records positional occupancy, where keys are positions and values represent the specific pieces occupying those positions.

This setup provides a detailed snapshot of the current board state, including both occupancy and potential interactions, which will serve as the baseline for subsequent analysis.

6.2.2 Phase B: Temporal Knowledge Graph Construction and Querying

In the second phase, we integrate the data from MoveRelationMap and StateMap into a temporal knowledge graph to capture dynamic and strategic relationships across both present and historical game states, using Neo4j. This knowledge graph models each piece as a node and each potential interaction (attack, protection, check, checkmate, discovered checks, and discovered attacks) as directed edges, enabling the following capabilities:

- Temporal Contextualization: A snapshot in the temporal graph is a representation of important information of each board state; preserving information about piece relationships and strategic positioning changes over time.
- Impact Propagation: Upon any positional change, a breadth-first search BFS traversal is initiated from the altered node, iterating across both current and historical graph states to identify and propagate any cascading influences. If a piece influences another, which in turn influences a third (whether in the present state or any prior game state), the system retains these relationships.

This BFS-based traversal identifies the depth of influence resulting from a positional change and gathers all directly and indirectly impacted nodes and edges for further analysis. The retrieved context is then passed along to Phase C, facilitating a nuanced commentary that can address both immediate and cumulative game-state changes.

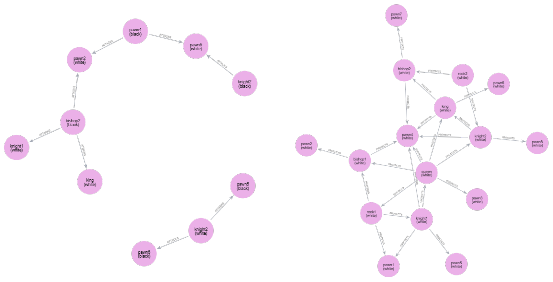


Figure 3: Snapshots of the knowledge graph for attack (left) and protect (right) configurations

6.2.3 Phase C: Contextual Language Generation

The final phase involves generating the commentary by integrating the insights derived from Phase A (current state analysis), Phase B (historical context) into a fine-tuned Large Language Model specifically trained on chess commentary data. Additional game engine data—like analysis of piece values, suggested moves, or evaluation scores—is provided to the LLM to generate commentary. Commentary generated with this data will be compared to commentary created without it to observe any differences in output.

The LLM is prompted to produce commentary that reflects the immediate and strategic implications of each move. This includes narrating shifts in control, identifying threats and protections, explaining the tactical relevance of discovered checks or attacks, and contextualizing current moves within the games broader progression. The generated output provides spectators with a layered understanding of the game, revealing both the visible and the underlying strategic considerations.

6.3 Baseline

We will compare our system against previous methods for automated chess commentary, specifically those proposed by (Zang et al., 2019), which integrates a neural chess engine with text generation, and (Lee et al., 2022), which utilizes advanced deep learning techniques for commentary. The comparison will be conducted using the same annotated chess games dataset to ensure consistency and fairness in the evaluation.

7 Experiments and Results

7.1 Initial Experiments

To evaluate the effectiveness of our approach for generating real-time chess commentary, we conducted a series of experiments using both fine-tuned and pre-trained models. Specifically, we explored three key approaches: zero-shot, few-shot, and sequential input strategies. For fine-tuning, we utilized two open-source models - Falcon 8B and Mistral 8B - to adapt them for the task of chess commentary generation. In the sequential input setup, multiple consecutive moves were provided as input to the model to enable richer context and improve the coherence of commentary.

Additionally, we experimented with a pre-trained language model, ChatCohere, without further fine-tuning. Notably, the pre-trained ChatCohere model consistently outperformed the fine-tuned Falcon and Mistral models across various evaluation metrics. The outputs from ChatCohere were more accurate, insightful, and stylistically closer to human-generated commentary, highlighting the importance of robust pre-training on large and diverse datasets for tasks requiring nuanced contextual understanding.

These results suggest that while fine-tuning open-source models can provide reasonable performance, leveraging highly-optimized pre-trained models offers a significant advantage in terms of both quality and contextual relevance.These results motivated us to proceed with a pre-trained model instead of fine-tuning smaller large language models. This decision was also driven by the limited computational resources available for training and fine-tuning these models effectively. Future work may focus on combining the benefits of fine-tuning with pre-trained model architectures to further enhance commentary generation.

7.2 Results

We evaluated the performance of our automated chess commentary generation models using both fine-tuned and pre-trained approaches. Three key strategies—zero-shot, few-shot, and sequential input—were explored to analyze their effectiveness in generating accurate and coherent commentary. We utilized the Falcon 8B and Mistral 8B models for fine-tuning, while ChatCohere (a pre-trained model) was used as a baseline without additional fine-tuning. BLEU Score Comparison. The evaluation metrics included BLEU and BLEU-2 scores, as shown in Table 2.

Features	**BLEU**	**BLEU-2**
COMB (M)	2.07	20.13
COMB (M+T)	2.43	25.37
COMB (M+T+S)	1.83	28.86
GAC (M)	1.69	20.66
GAC (M+T)	1.94	24.11
GAC (M+T+S)	2.02	24.70
CAT (M)	1.90	19.96
Mistral 8B	1.33	1.2
ChatCohere	1.16	1.08

Table 2: BLEU & BLEU-2 score comparison between old and our proposed methods

Among the fine-tuned models, the Mistral 8B achieved a BLEU score of 1.33 and BLEU-2 score of 1.20.

The pre-trained ChatCohere model, without any fine-tuning, performed slightly lower with a BLEU score of 1.16 and BLEU-2 score of 1.08. Notably, the COMB (M+T+S) configuration achieved the highest BLEU-2 score of 28.86, demonstrating the effectiveness of providing richer context through sequential inputs.

While the results show that fine-tuning provides reasonable performance improvements, the pre-trained ChatCohere model delivered superior stylistic quality and contextual relevance in generated commentary, as highlighted during qualitative analysis.

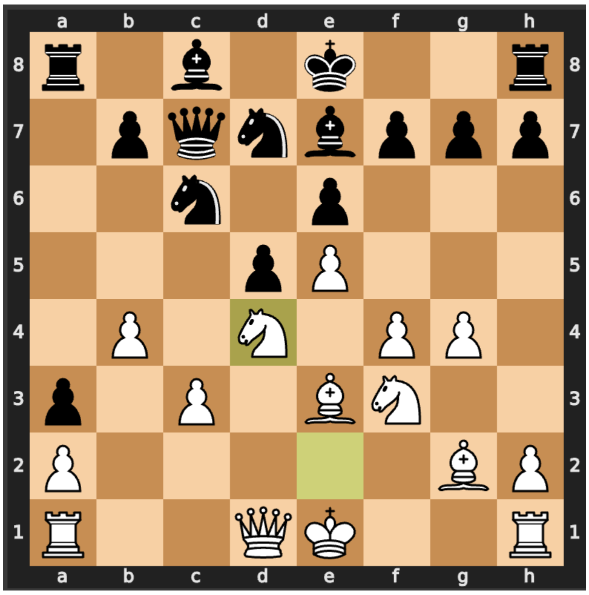


Figure 4: An example of the commentary generated: "Black's knight and bishop threaten White's center, with the knight also applying pressure to White's knight. White's knight1 has taken out a black pawn, but Black's pieces are well-protected. A cautious approach is needed."

The output shown in Figure 4 in demonstrates the ability of pre-trained models to deliver insightful and natural commentary without additional fine-tuning. The sequential input strategy, which provides a richer historical context for moves, further enhances the coherence and depth of the commentary.

The pre-trained ChatCohere model consistently outperformed the fine-tuned Falcon and Mistral models in generating commentary that was more accurate, nuanced, and stylistically aligned with human commentators.

7.3 Human Evaluation

To evaluate the quality of the generated chess commentary, we conducted a human evaluation study to gather subjective feedback on the commentary's effectiveness. For this purpose, we designed and distributed a survey that included a given game state (presented as a chessboard snapshot or move description) along with the corresponding commentary generated by our model. The survey aimed to assess key aspects of the commentary, such as accuracy (how closely the commentary aligns with the actual game state), coherence (how logically the commentary flows), informativeness (whether it provides meaningful insights), and relevance (how well it reflects the specific game situation).

Participants were asked to evaluate the commentary on a scale and provide qualitative feedback through open-ended questions. The survey questions were structured to capture multiple dimensions of commentary quality, ensuring a comprehensive evaluation. For instance, respondents were asked if the commentary accurately described the moves, whether it was easy to understand, and if it added value to their understanding of the game.

Once the responses were collected, we performed an analysis of the survey results. The findings indicated that the commentary generated by our model was generally well-received by participants. A significant portion of respondents noted that the model-produced commentary excelled at move descriptions and game insights, which are critical for engaging and informative chess analysis. Furthermore, many participants found the commentary to be coherent and relevant, particularly in describing the logical flow of moves and offering tactical observations.

Question	**Options**	**Results (n = 127)**
Fluency Scale	1	3.7%
	2	7.4%
	3	25.9%
	4	33.3%
	5	29.6%
Move Inference	Yes	74.1%
	No	25.9%
Correct Commentary	Yes	66.7%
	No	33.3%
Enhances Understanding of Game Strategy	Yes	59.3%
	No	40.7%

Table 3: Human evaluation study results for Automated Chess Commentary

However, there were occasional instances where participants observed that the commentary lacked the depth or nuance that might be expected from expert human commentary, particularly in highly complex game states. Despite this, the overall evaluation confirmed that the model performs well in delivering clear, accurate, and useful commentary for a wide range of chess positions.

The human evaluation study provided us with valuable insights into the strengths and limitations of our system. The encouraging feedback from participants validated the effectiveness of our approach and demonstrated the practical applicability of the generated commentary.

8 Future Work

Future work for this system includes expanding its framework - combining knowledge graphs, neural networks, and contextual language generation - to support other strategy-based games such as Go, Shogi, or modern board games like Settlers of Catan, which would involve fine-tuning models and designing game-specific knowledge graphs. Additionally, enhancing the system for real-time commentary during live chess games is a critical step, requiring optimization of the temporal knowledge graph and language model to ensure minimal latency while delivering high-quality insights. Personalization is another promising direction, where the model could adapt to different audience preferences by providing simplified explanations for beginners, deep strategic insights for advanced players, or entertainment-focused commentary for casual viewers.

9 Conclusion

The disparity between human and machine understanding of chess lies in their respective strengths and limitations. Modern chess engines and AI models have surpassed human players in their ability to calculate moves with extraordinary depth and speed. Machines can explore millions of potential move sequences in an instant, identify optimal moves, and foresee outcomes far beyond the scope of human calculation. This computational prowess enables machines to outperform even the best grandmasters in gameplay, as they operate purely on mathematical evaluation, optimization, and precision. However, despite their unmatched ability to play superior moves, machines often struggle to articulate the reasoning behind their decisions in a way that aligns with human understanding.

In contrast, humans possess an intuitive and pragmatic understanding of chess, developed through experience, pattern recognition, and strategic insight. While humans cannot match a machine's brute-force calculation, they excel at delivering commentary that explains the game's flow, the quality of moves, and the underlying ideas in natural language. Human commentary can weave together storytelling, emotional context, and qualitative judgments that resonate with an audience—providing an accessible interpretation of why certain moves are brilliant, risky, or flawed. Machines, on the other hand, lack this ability to express their decision-making process in human-like terms. They may execute a brilliant move, but without a mechanism to translate their computations into coherent natural language explanations, the move remains inscrutable to most observers.

This distinction highlights an important gap: while machines excel in move execution through pure calculation, they fall short in pragmatic understanding and communication—skills that are inherently human. Despite our efforts to bridge this gap using language models to generate commentary, the results we achieve still fall short of the depth, nuance, and creativity demonstrated by actual human commentators. Human commentators not only describe the moves but also weave in strategic insights, emotional context, and an engaging narrative that makes the game accessible and enjoyable for the audience. Our generated commentaries, while a step forward in combining machine precision with natural language generation, lack the subtlety and expertise that come from years of human experience and intuition in understanding and articulating the complexities of chess.

References

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, pages 128–135. ACM.

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.

Chloe Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339.

Remi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv preprint*, arXiv:1603.07771.

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.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 91–99. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint*, arXiv:1509.00838.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *arXiv preprint*, arXiv:1707.08052.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2016. Reference-aware language models. *arXiv preprint*, arXiv:1611.01628.

Hongyu Zang, Zhiwei Yu, and Xiaojun Wan. 2019. Automated chess commentator powered by neural chess engine. In *Annual Meeting of the Association for Computational Linguistics*.