This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International” license.

Stylometric Chess: the Styles of the Individual

John Walbran
walbr042@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

When analyzing work, it is often useful to focus on the author-specific aspects of work. The problem of stylometry concerns the identification of the author of a work based only on the characteristics of the work. With the advent of nerual networks, stylometric analysis can be effectively applied to more intricate work, and have been used to create chess engines that imitate individual players. This has many positive uses, but also raises serious privacy concerns.

Keywords: Stylometry, Neural Networks, Convolutional Neural Networks

1 Introduction

Stylometry is the study of the unique characteristics of specific authorship in work. This has historically been done primarily for the analysis of the authorship of text, such as the canonical works of Shakespeare [7]. With advancing technology, stylometric analysis has also been applied to other work. One use for stylometry in the modern era is in creating a chess model to imitate individual players. This has been done by a research group led by Reid Mcillroy-Young [5]. This paper aims to provide background and context in order to understand the function and implications of their model. Section 3 will start with an overview of neural networks, the core technology behind the chess model, focusing on a particular type of neural network, called a convolutional neural network. Section 4 discusses the model construction and efficacy. Then, section 5 discusses some of the ethical considerations of stylometry as a whole. Lastly, the paper concludes with a synopsis of the state of stylometric analysis.

2 Stylometry for Imitation Models

While stylometry was originally used only for the identification of the author of a work, modern neural networks enable the exploration of an extension of the original problem posed by stylometric analysis—imitation of an individual's behavior or authorship style. These imitation models, rather than attempt to identify the creator of work, attempt to synthesize work consistent with the creator's style. This is the more relevant form of stylometric model for a lot of the more contemporary work analyzed with stylometry, such as the work done by Gatys et al. which discusses neural networks to generate images in the style of famous painters [1].

3 Convolutional Neural Networks

Imagine you want to have a computer analyze a chess position. In order to do this, you need to have some way for the computer to understand the different features of the position, such as the pawn structure or the general piece placement, and then give it some context about what those features mean for the position. The most effective way to do this analysis is with a program called a convolutional neural network. Convolutional neural networks are a type of neural network that processes multi-dimensional data, like images. Neural networks are a predictive computational model that calibrate their predictions through an automated process of trial and error.

All neural networks consist of connected layers of nodes. Each connection between nodes has an associated weight, which controls how the values of each node influence the next layer in the network. Additionally, each node of a neural network may have an assigned activation function, which is a nonlinear function that is applied to each node before passing the value through the network. It is typical to assign each node in a layer the same activation function. Each activation function is nonlinear. This allows the neural network, which is otherwise a linear model, can extend to nonlinear behavior so it can effectively make predictions on nonlinear data. A neural network that contains only these interconnected layers is called a multi-layer perceptron (MLP). To use the network, an input is passed into the input layer, and then the input gets multiplied by the first set of weights, after which the first layers' activation function is applied to the output of the layer, which then is passed to the next layer in much the same manner. The process is then continued until the output layer is reached, at which point the neural network returns the values in the output layer.

When initially creating a neural network, weights are often randomly assigned, and then are refined automatically through a process called training. Training occurs over a dataset, called training data, which has the same form as the data that the neural network will eventually make predictions on, but also includes information about the result of that data, which allows the neural network to refine its predictions during training. This refinement requires a loss function, which is a quantitative representation of the error of a prediction by the model, which allows the weights to be adjusted to improve the guesses over time.

3.1 Convolutions

Convolutional neural networks (CNNs) are used when the input data is multi-dimensional. This is usually two dimensional data, like images. CNNs do this by analyzing the spatial relation between points in the input, as well as the values of the data. Convolutional neural networks achieve this processing through a collection of layers that are specific to convolutional neural networks: filter layers and pooling layers. Filter layers apply an operation called a convolution to the input. A convolution has a filter, which is a small matrix of weights that get applied iteratively to each region of the input data by taking the sum of elementwise multiplication. The results of these operations are then sent into an output array, which is passed into the next layer. An example of this is shown in figure 1. The weights in the filter are trained so that each different filter is reducing the region into the impact of a feature, such as edges or corners [3].

It is common for a neural network to have many filtering layers which extract different features from the input, so these different features are collected in pooling layers, and collects them into a summary of the region, which are then passed through the rest of the network.

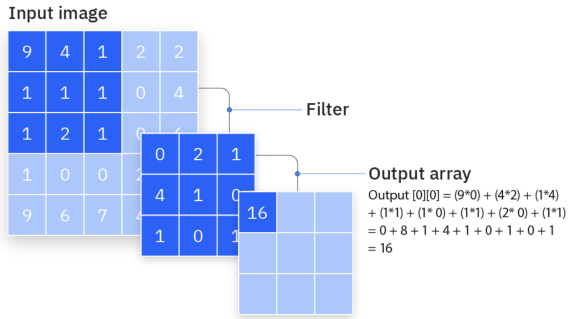

Figure 1. An example of a filter layer in a convolutional neural network.[3].

3.2 Residual Networks

Often in cases where many different CNN layers are applied to an image, the network can lose a sense of context within the larger scheme. In order to combat this, a slightly altered network architecture is used, called a residual convolutional neural network. A residual convolutional neural network, in addition to passing the outputs of the different convolutional layers forward in the network, it also passes forward the original image, providing more context to later stages of the network.

4 Case Study: Stylometric Chess Engine

Mcillroy-Young et al., as a continuation of their work on the Maia chess engine [4], created a model for stylometric analysis of chess players. The purpose of developing this model was to provide an individualized training tool for amateur players, who want to improve at the game for their own enjoyment, despite the fact that top computer engines have long since beaten the top human players [5]. Their most recent model boasts the ability to identify a specific player from a pool of candidates of 98% with about 100 labeled games for each player. The hope of the designers for this model is to provide individualized analysis of the strengths and weaknesses of an individual, which could then be refined to provide either personalized practice exercises, or more broadly, refinement and assistance for the individual [5].

4.1 Model Structure

Mcillroy-Young *et al.* created a model for performing stylometric analysis on chess games. Their model analyzes the core features of a game and extracts the essence of the game into a game vector. Game vectors consider the style of the game for one player of a game, rather than both sides simultaneously. A game vector is created by taking different aspects of the game over time, such as the pawn structure or whether a player has castled, and combining them into a much smaller number of axes. This allows for the exact determination of the position of the game within a multi-dimensional space. The different axes are automatically determined by the model, and are therefore not human readable. The spatial proximity of different game vectors represents similarity between different games. These game vectors can then be analyzed to identify individual players, by considering the proximity of a potential game to the game vector of games by that player. This can be done efficiently by looking at the cetroid $\vec{c}$ of the game vectors of the different games by that player.

$$\vec{c} = \left\langle \frac{\Sigma\vec{v}_1}{n}, \frac{\Sigma\vec{v}_2}{n}, \ldots, \frac{\Sigma\vec{v}_n}{n} \right\rangle$$

Each component of the centroid is the sum of the corresponding components of each point $\Sigma\vec{v}_i$, divided by the total number of points n. This gives the final centroid as the average of the coordinates of the input points. A visual example of the centroid of two dimensional points is given in figure 2.

The model takes a game as a sequence of moves. A move is stored as a 34-channel 8x8 grid, with one channel for each type of piece for each side, as well as storing game metadata, such as castling rights, draw by repetition, etc. [5]. A move is stored as two positions, representing the position before and after the move. The sequence of moves is passed in its entirety to the network as input. In order to encode the game, each move is passed through a residual CNN which creates a positional encoding for the move. Figure 3 shows this as a

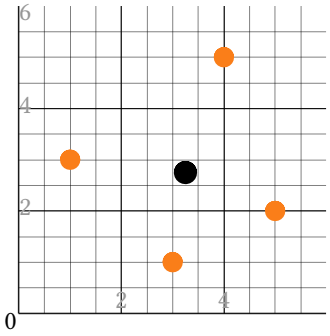

Figure 2. An example of a centroid of a set of points.

series of residual CNN blocks and a multi-layer perceptron step. This positional encoding is a representation of the relevant aspects of the position, such as pawn structure or piece placement. Each positional encoding is created separately, and the sequence of position encodings are passed through the model to be further refined into a game encoding. Once the moves of the game are converted into position encodings, they are then sent into a transformer. A transformer is a type of neural network that can take a sequence of inputs, and then extracts the essence of the positional encodings over the course of the game for one side of the game. This is seen in figure 3 as the transformer block, and a traditional MLP step. Finally, each value in the output is normalized, being mapped to a fixed set of ranges. The normalized output of the network is the new game vector.

4.2 Expanding to New Players

One of the main advantages of using game vectors as a way to identify players is that it is easily extendable to new players. The model created by Mcillroy-Young *et al.* identifies players based on the centroid of their game vectors. Because of this, and since the model creates analysis at a game level, not a player level, adding a new player to the model's known players only requires adding more games from that player. Mcillroy-Young *et al.* found that it takes 100 games from a player to create a representation of that player that preserves accuracy of predictions [5].

4.3 Training Data

Mcillroy-Young *et al.* trained their model on publicly available games through the lichess database [5]. They filtered their data to consider the games of players meeting the following criteria:

- Players who had played more than 1000 games on the platform.
- Players who were active as of December of 2020.
- Players whose ratings were between 1000 and 1900.
- Players with a low rating variance as of December 2020.

In addition to their main training data, they constructed a separate dataset that consisted of the games from master level players from multiple major online platforms [5]. After filtering players based on the above criteria, their main dataset had a pool of 41,184 different players, and a total of 67.5 million usable games. These players were then split into two sets of players, seen players, whose games were used for training the model, and unseen players, whose games were used to test the model. There were 16,181 different seen players, and 25,003 different unseen players. Furthermore, there were a total of 63.7 million seen games, and 4.98 million unseen games. These players were further sorted into different categories based on the number of games played. These buckets included 1,000-5,000 games, 5,000-10,000 games, and increasing buckets of 10,000 games up to the final bucket of 40,000 games or more. Games that were less than 10 moves long were discarded, since they would not be substantial enough to be meaningful in training the model. Additionally, the games from each player were split into training games, reference games, and query games. There were 100 reference games and 100 query games for each player, regardless of total games played. Additionally, each set of games for each player were distinct, with no game appearing in more than one set. The reference games were used to form a knowledge base for the model, and the query games were used as candidate games for identification [5].

When considering the master players, Mcillroy-Young *et al.* considered the top 1500 players from the leaderboards of the top online chess platforms, and filtered for players with at least 950 games on the platform, and games that were at least 10 moves long [5]. The dataset of master games was divided similarly to the main dataset, however, it was not divided into buckets based on number of games, but instead are considered completely. The reference and query datasets are created in the same manner as the main dataset.

4.4 Model Evaluation

Mcillroy-Young *et al.* evaluated their model by first creating a pool of candidate players with the reference games, and attempted to identify candidate players for the query games. Due to the construction, any candidate game is guaranteed to have been played by one of the known players from the reference pool. Additionally, they also decided to consider games from a variable starting point k, in order to determine the impact of different stages of the game on the identifiability of games. The model was evaluated both using the first 15 moves of the opening, and without. This was done mainly because the opening phase—which is usually the first 10 or so moves of a game—is often memorized, and most players play the same openings most games. This means that the opening is the most useful section of the game for determining the player of the game. This means that identifying the game style without the opening is a more important measure for the efficacy of the model.

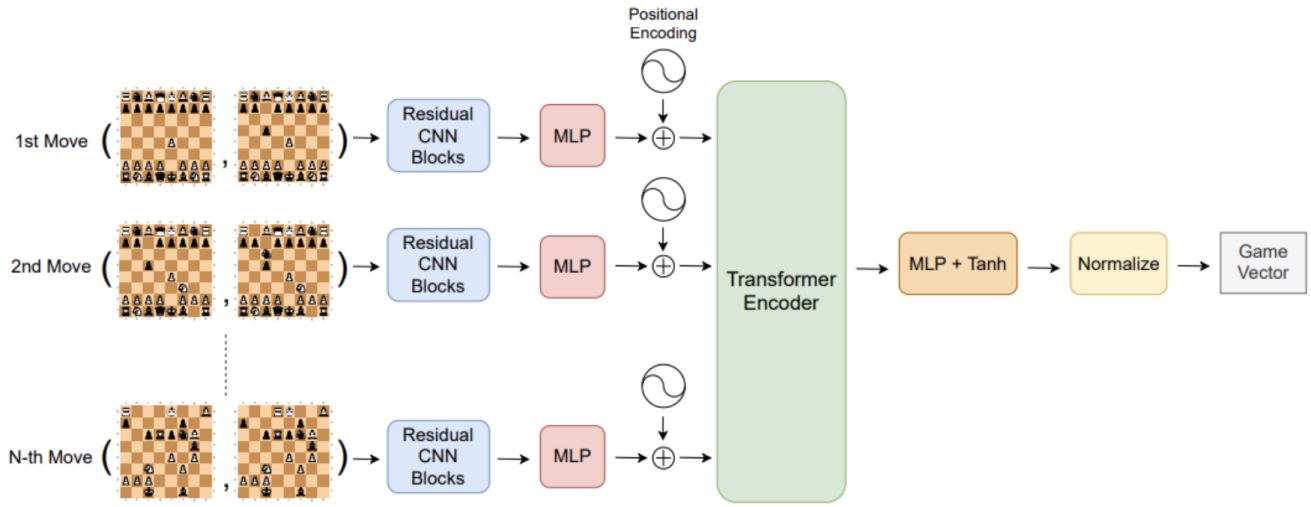

Figure 3. An overview of the neural network used for stylometric analysis of chess [5].

The model was trained using the seen player universe, considering all games in their entirety [5]. This created a space defined by the game vectors of each game by each player. Once this baseline is established, new players can be added to a candidate space by applying the model to create a new set of game vectors. Therefore, adding a new player to this set requires taking game vector representations of the games in their query set, creating a general representation using the centroid of those game vectors, and comparing that representation against the centroid representations of all the players in a candidate pool, returning the closest representation within the candidate pool. They created a similar candidate space for the high ranking players as well [5].

4.5 Results

The primary tests done by Mcillroy-Young *et al.* were on the main dataset, considering the amateur players with more than 10,000 games, and considering games from move 15 onward. They additionally trained a model on the master games. They then evaluated the performance of those models, the results of which can be seen in table 1. As can be seen from table 1, the best results were achieved when the candidate pool is smallest, seen by the tests of just the master games in the dedicated model, or when the model can consider the entire game for the amateur players. Of particular note in these results is that it's clear that knowledge of the opening has a substantial impact on the ability to identify a given player, as can be seen by the difference in amateur players with and without the opening. Model performance was measured as an accuracy, which is a value between 0 and 1 representing how often the model correctly identified the player for each game. The model had an accuracy of 0.854 without the opening, and a 0.982 accuracy with knowledge of the opening.

Of additional note is how the model performed with data points outside of distribution. In this case, considering the evaluation of the master games by the primary model, which was trained on amateur games. While the model did not perform particularly well at an accuracy of 0.308, especially when compared to the performance of the dedicated model, at 0.953, the performance of the model did not significantly diminish when master games were considered in addition to the amateur games. This consistency indicates that the master games are found in a different region of the game vector space as amateur games [5]. As a final analysis, Mcillroy-Young *et al.* found that even when considering master games from move 15 forward, there was tight clustering of games based on opening move. This demonstrates that, even without considering the actual opening moves, the choice of opening has a distinct impact on the character of the entire game.

Players Tested	Accuracy
Amateurs (10K+ games, move 15 onward)	0.854
Amateurs (10K+ games, whole game)	0.982
Amateurs (Similar ratings)	0.926
Amateurs (All games)	0.54
Masters (Dedicated model)	0.953
Masters (Primary model)	0.308
Masters $\bigcup$ Amateurs	0.301

Table 1. The results from Mcillroy-Young *et al.* [5]

5 Ethical Considerations of Stylometry

With the research done by Mcillroy-Young *et al.* possibly having serious privacy ramifications, NeurIPS, their publishers, requested that Mcillroy-Young *et al.* also published a companion paper discussing the ethical implications of accurate stylometry [2, 6]. Their paper, *Mimetic Models: Ethical Implications of AI that Acts like You*, addresses the ethical implications of mimetic models—imitation stylometry models—through a series of case studies, demonstrating the benefits and pitfalls of various situations.

The main pitfalls that Mcillroy-Young *et al.* identify in their analysis are removal of privacy, and the devaluing of human creations. Removal of privacy is quite apparent when considering models that are trained for the most basic stylometric analysis—identifying the author of an anonymous text. This is useful for historical analysis, allowing for recovery of information that is otherwise unattainable, however, in a modern context, it can also reveal people who otherwise want to remain anonymous.

Regarding the second concern, devaluing the creations of humans, Mcillroy-Young *et al.* look at a hypothetical imitation model of a teacher, or even a teacher assistant [6]. Consider an imitation model of a teacher. This model could be accessible remotely to answer questions about work outside of predetermined class time or office hours, which allows the teacher to have better separation of their occupation and the rest of time. Additionally, this imitation model could help to write lesson plans, or to help give feedback on student work. This all saves time and energy for the teacher, allowing them to spend their time on refinement and enrichment of the curriculum, or spend that time on pursuits outside their occupation. The downside of this is that eventually, if the imitation model starts to be competent at enough aspects of the work, that there would eventually be a significant devaluation of the work actually done by the human teacher.

6 Conclusion

Machine learning allows for much more comprehensive stylometric analysis than was previously possible. This not only comes in the form of deeper analysis of text, but also allows us to analyze and quantify individual style in more sophisticated work. Chess is a rich domain to explore due to its continued popularity, and the wealth of historical usage of computers for its analysis. Stylometric analysis in chess requires the construction of an embedding space for chess games. This embedding space can be used to identify players, with the hopes of further analyzing chess players at different strengths, as well as to create personalized chess training tools. While there are many positive aspects of using stylometric models, there are concerns and pitfalls to avoid as well. Stylometric models can be used to significant benefits when used responsibly, however, they have the potential to cause serious harm when used maliciously.

Acknowledgments

I would like to thank Professor Elena Machkasova for advising during the writing process, as well as Professor Wenkai Guan for persistent interest and advice throughout the writing of this paper. Additionally, I'd like to thank Josh Eklund for his excellent review of this paper.

References

[1] Gatys, Ecker, and Bethge. 2015. A Neural Algorithm of Artistic Style. https://arxiv.org/abs/1508.06576

[2] Hutson. 2022. AI unmasks anonymous chess players, posing privacy risks. https://www.science.org/content/article/ai-unmasks-anonymous-chess-players-posing-privacy-risks

[3] IBM. [n. d.]. What are convolutional neural networks? https://www.ibm.com/topics/convolutional-neural-networks

[4] McIlroy-Young, Sen, Kleinberg, and Anderson. 2020. Aligning Superhuman AI with Human Behavior: Chess as a Model System. *ACM SIGKDD* (2020). https://doi.org/10.48550/arXiv.2006.01855

[5] McIlroy-Young, Wang, Sen, Kleinberg, and Anderson. 2021. Detecting Individual Decision-Making Style: Exploring Behavioral Stylometry in Chess. *NeurIPS 2021* 34 (2021), 23 pages. https://doi.org/10.48550/arXiv.2208.01366

[6] McIlroy-Young, Wang, Sen, Kleinberg, and Anderson. 2022. Mimetic Models: Ethical Implications of AI that Acts Like You. https://doi.org/10.48550/arXiv.2207.09394

[7] Wikipedia. [n. d.]. Shakespeare authorship question. https://en.wikipedia.org/wiki/Shakespeare_authorship_question#:~:text=The%20historical%20record%20is%20unequivocal,of%20William%20Shakespeare%20of%20Stratford.