The Dots and Boxes Records Storing Standard Format for Machine Learning and The Design and Implementation of Its Generation Tool

Ming Gao¹, Shuqin Li¹*, Meng Ding¹ and Meng Kun¹

¹School of Computer Science, Beijing Information Science & Technology University, Beijing, China

*Corresponding author e-mail: lishuqin_de@126.com

Abstract. Dots and Boxes is a very popular game for two players. There are many teams in the world who are studying smart game programs about this game. Unfortunately, different from the popular game Go, there are no uniform Dots and Boxes game storing records standards format. This results in a large number of classical match records that cannot be preserved, and it also affects the study of this game algorithms. In particular, machine learning algorithms rely on a large number of records. The AlphaGo developed by Google defeated human top players for the first time in Go with the help of a large number of human chess games. In order to facilitate the recording and saving of Dots and Boxes game records. Based on the JSON standard, this paper designs a Dots and Boxes game records standard format and its storage standard, which has excellent expansion and strong compatibility. It also supports cross-platform applications, facilitates secondary development, and is easy to read and write. In order to facilitate the use and promotion of this standard format, simplify the development process of the game AI algorithm. This article uses Python3 language, based on PyQt5 framework, to implement a Dots and Boxes game software DotsAndBoxes, which can generate and read the above-mentioned game records standard file. It has the characteristics of strong versatility, high portability, good scalability, and low maintenance cost.

1. Introduction

Game manual is a record of the game process, and it is also an important material for analyzing and researching the game process. As with all human knowledge, the skill of chess games requires long-term accumulation, and the basis for accumulation is the record of the game process. In 2016, the AlphaGo developed by Google[1] defeated human top players for the first time in Go with the help of a large number of human chess games. The subsequent AlphaGo Zero[2] did not use human match data, but it still required self-play to produce the game records. It can be seen from this that the importance of game manual for the study of board games.

Dots and Boxes is a very popular game for two players[3]. Internationally, it began very early to study smart game programs about this game. The research in China on the Dots and Boxes game started late. In recent years, under the promotion of the University Computer Games Championship, more and more colleges and universities have joined the study of Dots and Boxes game program. Unfortunately, different from the popular game Go, there are no uniform Dots and Boxes game storing
records standards format so far. Even the chessboard of this game has no coordinate mark method. This not only caused the game process cannot be saved effectively, but also caused difficulties for the retention of the classical chess game spectrum of the Dots and Boxes game, and thus affected the development of this algorithm. Therefore, it is very necessary and urgent to design a record storing standard format of Dots and Boxes game.

In order to facilitate the recording and saving of Dots and Boxes game records, and also to facilitate the study of deep learning in this game. Based on the JSON standard, this paper designs a Dots and Boxes game records standard format and its storage standard, which has excellent expansion and strong compatibility. It also supports cross platform applications, facilitates secondary development, and is easy to read and write.

In order to facilitate the use and promotion of the Dots and Boxes game records standard format and to simplify the development process of this game algorithm and reduce game AI development costs. This article uses Python3 language, based on PyQt5 framework, to implement a Dots and Boxes game software: DotsAndBoxes. It can generate and read the game records standard format file. In addition to realizing the game function of human, the game software also has an AI programming API for the Dots and Boxes game, which can easily load custom game AIs to achieve man-machine game and machine-machine game functions. It has the characteristics of strong versatility, high portability, good scalability, and low maintenance cost. And has been released to PyPI, any computer installed Python3 operating environment can be very easy to install and use.

2. The design of Dots and Boxes records storing standard format

2.1. The purpose of the game manual

The game records storing standard format is essentially a computer data storage format. To design a reasonable and efficient game records format, we must clearly define the requirements. The main application purposes of the game records are the following: Completely record the game process for future reference, analysis and research. Save the residuals so that you can continue playing games later, or analyses and study the endgame itself. Batch storage of a large number of game records for use in deep learning algorithms.

2.2. The design of file basic storage

Compared to binary storage, plain text storage is more suitable for the game record files. Because binary stored game files must be parsed by software before they can be read by humans, this is obviously not conducive to manually checking the game record in special circumstances. At the same time, the plain text storage method can facilitate the spread and sharing of the game. The users can easily send and publish their own game records through e-mail, instant messaging software, or websites. All users can use the basic text editor to view. It facilitates the popularization of the standard format.

In order to better achieve full-platform compatibility, the Dots and Boxes records storing standard format uses UTF-8 (Unicode Transformation Format (8-bit) encoding). UTF-8 encoding is currently the most widely used text encoding standard in the world[4]. As of May 2018 accounts for 91.5% of all web pages use UTF-8 encoding[5]. The unified code allows the game records to be easily used by users all over the world without the problem of parsing exceptions. At the same time, UTF-8 encoding is also compatible with ASCII encoding. The first 128 characters of UTF-8 encoding are completely the same as the ASCII encoding[6]. And these 128 characters can cover most of the characters in the game record file except the team name. Good compatibility guarantees that even if used on low-performance or older devices, the game record file can guarantee usability.

2.3. The design of coordinate system for chessboard

The Dots and Boxes game chessboard is a 6*6 dot matrix[7]. Starting with an empty grid of dots, a chesspiece is a single horizontal or vertical line between two un-joined adjacent dots. Two players take turns adding chesspiece, one at a time. The player who completes the fourth side of a 1×1 box earns
one point and adds another chess piece. The bottom left corner of the chess-board coordinates is the origin, and the letters from left to right are ABCDEF, and the numbers from bottom to top are 123456.

2.4. The design of coordinate system for chesspiece and moves
For the position of a chesspiece, use a doublet. The first item is the starting point coordinate (for the horizontal chesspiece, the left end point is the starting point; for the vertical chesspiece, the lower end point is the starting point), and the starting point coordinate is represented by <alphanumeric>, such as "B4". The second item is the type of chesspiece, with the horizontal being "H" and the longitudinal being "V". Chesspiece positions are enclosed in English parentheses, separated by a comma ",". The place of red chesspiece in Figure 1 is represented by "(B4,V)".

In front of each chesspiece, the player to which the chesspiece belongs is marked with red for "R" and blue for "B". As in the red chess piece in Figure 1, the complete chesspiece is represented as "R(B4,V)". The chess-piece coordinates are not case sensitive, i.e. "R(B4,V)" and "r(b4,v)" are the same chesspiece.

With the starting point + piece type description, all the chess pieces in the entire chessboard can be uniquely marked. Some traditional Dots and Boxes game description methods use the grid coordinates plus the position of the chesspieces to describe the chesspieces, as shown in Figure 2. In this way of representation, the red chess-piece in Figure 2 has two coordinates: "A4 right" or "B4 left". Although this representation is more intuitive and easier to use in the unique long-chain structure of the Dots and Boxes game, it can cause difficulties in computer analysis, and people are prone to ambiguity when they directly read the game records. Because there are two kinds of descriptions for a chesspiece, the computer needs to write a lot of logic to determine whether there are repeated cases or not during the process of resolving the game. It checks that the repetitive calculation algorithm consumes at least twice as much computing resources as the unique piece description inspection algorithm.

2.5. The design of json-based game information format
For each game manual, it should contain the following data: Red player name or ID, Blue player name or ID, Date of playing, Game event/location, Winner, Score, Game process.

Most of the above information is structured data. In order to facilitate human processing while facilitating machine processing, the game manual is stored in JSON format. JSON derived from JavaScript uses a text format that is completely independent of the programming language to represent data. The concise and clear hierarchy makes JSON an ideal data exchange language. It is easy for people to read, but also easy to machine analysis and generation, and effectively improve the network transmission efficiency.

XML is more suitable for storing structured data than JSON. However, XML is too complex to represent data with a simple structure[8]. The same game record using XML format description is about 5% larger than using JSON format. At the same time, XML has a large number of additional specifications, such as DTD, XSD, XPath, XSLT, etc., in order to be able to handle more complex data structures and larger data volumes. Therefore, tools or programming libraries that support XML processing are generally more complex. Because JSON does not support complex operations, its processing and parsing tools are more streamlined and faster[9].
Therefore, this article determines that the Dots and Boxes records storing standard format is stored in JSON format. In the JSON specification, all data structures are unified into dictionaries and arrays. Each game in the records file is a dictionary. The game dictionary contains 3 key-value pairs, which are: "R": "team A" (red party name/ID), "B": "team B" (blue party name/ID), "winner": "R", "RScore": "4" (Red score), "BScore": "9" (Blue score), "Date": "2017-07-30", "Event": "2017 CCGC" and "game": "The game process string"

In the game process string, the coordinate representation of pieces is generated according to the order of the falling pieces. The coordinates of each piece are separated by a semicolon ";".

2.6. The extended and compatible
Using structured JSON descriptions can guarantee excellent extensibility and compatibility of the Dots and Boxes records storing standard format. In terms of extensibility, the standard format only specifies which key-value pairs have to include, and the rest can be expanded by the user according to the situation. At the same time, good extensibility allows incremental upgrades for each version of the game standard format. The new version of the game standard format can be used to achieve excellent forward compatibility and reduce the cost of standard upgrades.

3. The dots and boxes game software design and implementation

3.1. Software demand analysis
If only design and publish the standard format, and do not provide complete tools for people use, it is very likely that the standard format will become a piece of paper. In order to facilitate the promotion of the Dots and Boxes records storing standard format and to popularize the Dots and Boxes game, it is necessary to develop a Dots and Boxes game software with good user interaction.

The function of the software is mainly divided into two levels: the first level is the good playing Dots and Boxes game function of human-computer interaction, and the second is the ability to generate and analyses the Dots and Boxes records storing standard format files, and reviews the game process of the Dots and Boxes game. The process facilitates the need for reconciliation and inspection after the game.

In the Dots and Boxes game function, the following function points are specifically included: Intuitive graphical chessboard. Click on an edge or connection point to achieve a move. The pieces are displayed with the corresponding player color. Previous step special mark. The enclosed grid is marked with the color of the player occupying the grid. Which player need adding the next piece.

Figure 3. Comparison of stored in the XML format and stored in JSON format
Show current score. Back/Restore a move. Judging winning and losing and giving tips. Save the game. Recovery game.

As for the process of recording the game between the two sides of the checkerboard game, and facilitating the need for reconciliation and inspection after the match, the following function points should be expanded: Export the Dots and Boxes records storing standard format files. Import a Dots and Boxes records storing standard format files. List all the chesspieces in order. Jump to any history step, including forward jump and backward jump. When you do not change the position of history, the historical information after the current step will not be cleared.

3.2. Software design
In order to meet the requirements of compatibility, portability, extensibility and ease of maintenance, the software uses Python 3 as the main development language. At the same time using PyQt5 as graphical interface library to achieve graphical interaction.

In this paper, the Dots and Boxes game software is divided into two main parts, one part is the pure Dots and Boxes game logic module without a graphical interface, and the other part is a graphical human-machine interface module.

3.3. Detailed Design of Game Function Logic Module
The Dots and Boxes game logic module can operate as an independent console program, has the full Dots and Boxes game function, and reserves an interface for the game AI program to access. The graphical human-computer interaction interface module only includes the UI code and the interface logic control code and realizes the graphical game playing function by calling the Dots and Boxes game logic module.

The game function logic module mainly includes three parts: control function, game function and other functions. Control features include starting/ending games, saving/loading games, loading and modifying players. The game features include falling, repenting, and jumping to any step-in history. Additional features are loading and exporting the Dots and Boxes records storing standard format files. There are two main types of this module: DotsAndBoxes class, Game class. The DotsAndBoxes class implements the checkerboard game logic, and the Game class implements the logic of a grid chess game. The main difference between the two classes is that the former runs through the entire game life cycle, may include multiple game playing process, even contains more than 2 players, because some players may be replaced after several sets of playing matches, while the latter only exist in one inventory. There are only two players in the chess game process, and there is no logic for player replacement.

The DotsAndBoxes class is the top-level class in the entire module and needs to implement all functional interfaces. The DotsAndBoxes class only needs to instantiate an object throughout the entire software runtime. The DotsAndBoxes class contains a Game attribute, two protected Player attributes, and a read-only history attribute. The Player property can only be assigned if there is no unfinished Game instance (i.e., no game has been started or the game has already been won). The virtual attribute current_step can be used to obtain the current number of steps, and the virtual attribute current_player can be used to obtain the current player. The DotsAndBoxes class provides for starting the game, ending the game, falling, repenting, and jumping methods. There are two methods of falling the stack, one using a string coordinate parameter and the other using a chess object parameter. Saving/loading games and loading and exporting standard game features can be combined into two methods, load_from_file and save_to_file, to differentiate between game archiving and standard format files by passing in different parameters.

The Game class is a game class that contains two read-only Player properties, a read-only history property, and a read-only board property. A virtual attribute score is used to obtain the score, a virtual attribute current_player_color attribute is used to obtain the current player color, a virtual attribute is_end is used to determine whether to end, a virtual attribute winner is used to obtain the winner. Provides ways to move and back.

The history list of pieces attribute in the Game object is a real-time change, that is, when the player chooses back, the history of the last step will be destroyed. However, the history list attribute in the
DotsAndBoxes object is a delayed change. When the player chooses back, the chesspiece is not deleted, but the current step pointer is used to move forward. The reason for this design is that the revocation of repentance can be realized, and it can also be understood as if it is not changed. History can be arbitrarily jumped in historical steps. Only when the position of the new chesspiece after the repentance game is different from the history (that is, when the history needs to be modified), the history of this step and all subsequent steps in the history is deleted, and the new chesspiece is added to the history list. Specific state transition see Figure 4.

4. Software testing and publishing

After the development was completed, we systematically tested the software running on different platforms. Figure 5 shows how the software works in the Windows 10 and Ubuntu Desktop 1604 systems.

To make it easier for users to use the game software, we use the setup tool in python to package the software and publish it to the official PyPI software source recommended by python[10]. Users can use "pip install DotsAndBoxes" for quick installation.

Acknowledgments

This work is Supported by National Natural Science Foundation of China (No.61502039), Supported by the special bidding project of teaching & education reform (2017JGZB08), and Supported by 2018 Beijing Information Science and Technology University Graduate Student Science and Technology Innovation Project.

References

[1] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.
[2] D Silver, J Schrittwieser, K Simonyan, et al. Mastering the game of Go without human knowledge.[J]. Nature, 2017, 550(7676):354.

[3] ‘Wikipedia contributors. Dots and Boxes. Wikipedia, The Free Encyclopedia. May 16, 2018, 15:03 UTC.’, https://en.wikipedia.org/w/index.php?title=Dots_and_Boxes&oldid=841554530. Accessed May 19, 2018.

[4] ‘Wikipedia contributors, "UTF-8," Wikipedia, The Free Encyclopedia’, https://en.wikipedia.org/w/index.php?title=UTF-8&oldid=841567320. Accessed May 20, 2018.

[5] ‘Usage Survey of Character Encodings broken down by Ranking’. w3techs.com. https://w3techs.com/technologies/cross/character_encoding/ranking. Retrieved 2018-05-16.

[6] Yergeau F. UTF-8, a transformation format of ISO 10646[J]. 2003.

[7] ‘The rules of the Dots and Boxes’. University Computer Games Championship. http://computergames.caaicn/jsgz02.html, Retrieved 2018-05-20.

[8] Wang G. Improving data transmission in web applications via the translation between XML and JSON[C]//Communications and Mobile Computing (CMC), 2011 Third International Conference on. IEEE, 2011: 182-185.

[9] H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representation,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS ’07), pp. 57-64, Apr. 2007, doi:10.1109/SCIS.2007.367670. (Conference proceedings)

[10] ‘DotsAndBoxes 1.0.3 Project description ’. pypi . org. https://pypi.org/project/DotsAndBoxes/. Retrieved 2018-05-20.