Lab. OiMW – team project (3-4 persons)

Specification version: December 11, 2019

# Main idea

The program is capable of analyzing, with a help of a chess engine, a series of chess games from a chess database written to a PGN-compatible file, looking for strong, non-trivial moves. For each game, it checks all positions occurring on the main line (skipping variations), to see if they meet certain criteria (filters). For each position where the best move suggested by the engine matches all the criteria, the program puts in the output PGN file a new chess game, containing at least a FEN header, defining starting position, and (at least) two best moves as indicated by the engine. The user should be able to configure search criteria, chess engine options, and format of the output PGN file.

# Synopsis

**strong-moves-extract** -h <> -cp <> -d <> -n <> -e <uci-server-configuration-file-path> <input-PGN-file-path> <output-PGN-file-path>

# Parameters

-h: what headers should be put to output PGN file (all, concise, minimal); default: minimal; meaning:

|  |  |  |
| --- | --- | --- |
| [Event "12th World Teams 2019"]  [Site "Astana KAZ"]  [Date "2019.03.12"]  [Round "7.1"]  [White "Swiercz,D"]  [Black "Nepomniachtchi,I"]  [Result "1/2-1/2"]  [WhiteTitle "GM"]  [BlackTitle "GM"]  [WhiteElo "2655"]  [BlackElo "2771"]  [ECO "D90"]  [Opening "Gruenfeld"]  [EventDate "2019.03.05"] | all: | all headers from the left +  [FEN <extracted position in FEN format>] |
| concise: | [White "Swiercz,D"]  [Black "Nepomniachtchi,I"]  [Site "Astana KAZ"]  [Date "2019.03.12"]  [FEN <extracted position in FEN format>] |
| minimal: | [FEN <extracted position in FEN format>] |

-cp: min. required cp (centipawns) difference between best and second best move shown by the engine; default: 50

-d: min engine search depth for best and second best move shown by the engine (in multivariation mode); default: 30

-n: number of variations in multi-variation mode; default: 2 (minimal acceptable value)

-e: path to UCI Server configuration file (relative or absolute); default: uciServer.json

Paths to input/output PGN files may be relative or absolute.

# Output PGN file:

For each position extracted from input PGN file, meeting all criteria, output PGN file should contain a single game entry composed of:

1. headers (including FEN), depending on the value given after -h switch
2. best move + position evaluation after this move as a comment + additional{G} comment if this move was actually played in the game
3. second best move as variation + position evaluation after this move as a comment + {G} comment if this move was actually played in the game
4. if the value given after -n switch is greater than 2, then for each next best move according to the engine the file contains this move + position evaluation after this move as a comment + {G} comment if this move was actually played in the game

**Remark**: all evaluations should be given from White’s perspective; e.g., +100 means that White has an advantage of one pawn, while -250 means that Black has an advantage of two and a half pawns.

# Position filters (encoding chess domain knowledge)

Only positions for which the best engine move matches all the filters should be put into the output PGN file. For each position in the input PGN file, apart from the main filter concerning min. cp difference between the best move and the second best move shown by the engine (run in MultiPV mode), also these filters should be jointly applied to the best move (the following list contains some suggestions, that SHOULD BE EXTENDED by at least one more filter that contributes to improvement of non-trivial strong move detection quality):

* the best move is not a simple (re-)capture of a piece with value missing for material equality
* the best move is not just a capture by a minor piece, leading to material advantage

Ideally (to support future program development), each filter should be an instance of interface MoveFilter defining just one method match with the following signature:

public boolean match(String FEN, String move);

where move is written down according to UCI notation (<initial-field><target-field>), e.g., g1f3 or a8a1. Return value true means that the filter matches the given move, which means that the move “survives” application of the filter (and should be further considered). Return value false means that the move does not fulfill the criteria checked by the filter, and thus the whole position for which that move was tested should not enter the output PGN file.

If the above solution with MoveFilter interface occurs to be hard to implement, as a minimal requirement working principle of each applied filter (apart from the main one concerning diff. in evaluations) should be documented in separate **filters.md** file.

# Program requirements

* Written in Java
* Communicates with chess engine using UCI Server (link given below), over the network (which enables to deploy the engine on another, possibly stronger machine); it is assumed that UCI Server configuration details are read on program startup from a JSON file
* Input and output PGN files are read from / saved to local machine
* PGN files are parsed using Java PGN reader (link given below)
* Distribution jar should be build from sources using Maven or Gradle
* Program source code should be **documented** (Javadoc) and submitted along with filters.md file
* Apache 2.0 License

# UCI server configuration file – example (uciServer.json)

{

"url": "http://127.0.0.1:8080",

"login": "test",

"password": "111111",

"engine": {

"name": "Stockfish 10 x64",

"options": [

"Threads":"4",

"Hash":"1024"

]

}

}

Array options should contain 0+ "parameter-name":"parameter value" pairs. E.g., if chosen engine responds to the uci command on its standard input with

option name Threads type spin default 1 min 1 max 512

then one can set its Threads option this way:

"Threads":"4"

which the program internally should realize by sending to the UCI server, once Web Socket connection is set up, the following command:

setoption name Threads value 4

**Remark**: engine’s option MultiPV should be ignored if set by the user – it should be set automatically based on the value after -n switch.

# References/specifications:

1. Portable Game Notation (PGN):
   1. <https://www.chessclub.com/help/PGN-spec>
   2. <http://www.saremba.de/chessgml/standards/pgn/pgn-complete.htm>
   3. potentially useful processor of PGN files: *pgn-extract* (<https://www.cs.kent.ac.uk/people/staff/djb/pgn-extract>)
   4. **Java PGN reader**: <http://www.cs.put.poznan.pl/mszelag/Software/software.html> – section **Chess-related software / Java PGN reader**
   5. Exemplary larger PGN file: <https://idss-msz.cs.put.poznan.pl/strong-moves-extract/twic1271.pgn>
   6. Numeric Annotation Glyphs (NAGs): <https://en.wikipedia.org/wiki/Numeric_Annotation_Glyphs>
2. Forsyth-Edwards Notation (FEN), Extended Position Description (EPD):
   1. [http://www.saremba.de/chessgml/standards/pgn/pgn- complete.htm#c16.1](http://www.saremba.de/chessgml/standards/pgn/pgn-%20complete.htm#c16.1)
   2. <https://pl.wikipedia.org/wiki/Notacja_Forsytha-Edwardsa>
3. Universal Chess Interface (UCI):
   1. <http://download.shredderchess.com/div/uci.zip>
   2. **UCI server**: <http://www.cs.put.poznan.pl/mszelag/Software/software.html> – section **Chess-related software / UCI server**
4. UCI & PGN compatible free chess GUI - SCID: <http://scid.sourceforge.net/>

# Chess engine

Any UCI-compatible chess engine wrapped by UCI server, e.g., [Stockfish 10](https://stockfishchess.org/) or [Leela Chess Zero](https://github.com/LeelaChessZero/lc0).

# Examples

## Exemplary input PGN file (just two games, one with comments, variations, and NAGs, all of which are handled by Java PGN reader):

[Event "12th World Teams 2019"]

[Site "Astana KAZ"]

[Date "2019.03.12"]

[Round "7.1"]

[White "Swiercz,D"]

[Black "Nepomniachtchi,I"]

[Result "1/2-1/2"]

[ECO "D90"]

1.d4 Nf6 2.c4 g6 3.Nc3 d5 {Gruenfeld defence} 4.Nf3 Bg7 5.h4

$5 c6 6.cxd5 cxd5 7.Bf4 ( 7.h5 Nxh5 ) O-O 8.e3 Nc6 $13 9.Qb3 Na5

10.Qa3 Nc6 11.Qb3 Na5 12.Qa3 Nc6 13.Bb5 Bd7 14.O-O a6 15.Be2

b5 16.Qb3 Na5 17.Qd1 {+0.48} Bg4 18.Ne5 Bxe2 19.Qxe2 e6

( 19...Nc4 20.Nxc4 bxc4 ( 20...dxc4 21.b3 b4 ) 21.b3 ) 20.Nd3

Qe7 21.Rac1 Rfc8 22.Bg5 Qf8 23.Nc5 Nc4 24.Rc2 Qe8 25.Rfc1 Nd7

26.Nd3 Ndb6 27.Bf4 h6 28.Nb1 Bf8 29.b3 Nd6 30.Nd2 Rxc2 31.Rxc2

Rc8 32.Qd1 Qd8 33.Bxd6 Bxd6 34.Nf3 Rc7 1/2-1/2

[Event "12th World Teams 2019"]

[Site "Astana KAZ"]

[Date "2019.03.12"]

[Round "7.1"]

[White "Grischuk,A"]

[Black "Sevian,Samuel"]

[Result "1/2-1/2"]

[ECO "D30"]

1. d4 Nf6 2. c4 e6 3. Nf3 d5 4. Bg5 h6 5. Bxf6 Qxf6 6. e3 Nd7 7. Nc3 c6 8. Bd3

g6 9. O-O Bg7 10. Re1 dxc4 11. Bxc4 O-O 12. e4 e5 13. d5 Nb6 14. Bb3 Rd8 15. h3

Bf8 16. Qe2 Bd7 17. Red1 Rac8 18. a4 a5 19. dxc6 Bxc6 20. Qe3 Nd7 21. Nd5 Bxd5

22. Rxd5 Nc5 23. Bc4 Rxd5 24. Bxd5 b6 25. Rc1 Kg7 26. g3 Rc7 27. Kg2 Bd6 28. b3

Na6 29. Rxc7 Bxc7 30. Qc3 Nc5 31. h4 Qd6 32. Qc4 Qe7 33. h5 Bd8 34. Qc3 Nd7 35.

hxg6 fxg6 36. Bc6 Qc5 37. Qc4 Qxc4 38. bxc4 Nb8 39. Bb5 Kf6 40. Ne1 Bc7 41. Nd3

g5 42. c5 bxc5 43. Nxc5 Bb6 44. Nd3 g4 45. Nb2 h5 46. Nc4 Bc7 47. Ne3 Kg5 48. f3

gxf3+ 49. Kxf3 Bb6 50. Nc4 Bc7 51. Ne3 Bb6 52. Nc4 Bc7 53. Kg2 h4 54. gxh4+ Kxh4

55. Kf3 Kg5 56. Ne3 Bb6 57. Nd5 Bc5 58. Ke2 Kg4 59. Nc7 Bg1 60. Ne6 Bb6 61. Kd3

Kf3 62. Nf8 Bd8 63. Ne6 Bb6 64. Ng7 Ba7 65. Nf5 Bf2 66. Nd6 Be1 67. Ne8 Bb4 68.

Ng7 Be1 69. Nf5 Bf2 70. Nd6 Be1 71. Nc4 Kf4 72. Nb6 Kf3 73. Nd5 Bh4 74. Ne3 Be1

75. Nc2 Bf2 76. Na3 Be1 77. Nc4 Kf4 78. Nd6 1/2-1/2

## Exemplary FENs (3 positions):

FEN for starting position:

rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1

After 1. e4:

rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1

After 1. ... c5:

rnbqkbnr/pp1ppppp/8/2p5/4P3/8/PPPP1PPP/RNBQKBNR w KQkq c6 0 2

After 2. Nf3:

rnbqkbnr/pp1ppppp/8/2p5/4P3/5N2/PPPP1PPP/RNBQKB1R b KQkq - 1 2

## Exemplary program run

?> **strong-moves-extract** -h minimal -cp 100 -d 36 -n 3 -e uciServer.json test.pgn test-positions.pgn

#search for positions in input file test.pgn, with minimal difference between the best and second best move suggested by the engine configured in uciServer.json, at depth 36 halfmoves, equal to 100 centipawns; for each position matching simultaneously all criteria (implemented filters), put in the output file test-positions.pgn game entry with minimal header (containing only FEN), best move on the main line, and second as well as third best moves as variations; attach a comment to each of these moves showing its engine evaluation; mark actual game move by {G} comment

Resulting PGN file test-positions.pgn (fragment):

[FEN "8/p5Q1/2ppq2p/3n1ppk/3B4/2P2P1P/P5P1/6K1 w - - 3 46"]

46. g4 {+29900} (46. Qxa7 {+0}{G}) (46. c4 {-51}) \*