Learning Chess Blindfolded: Evaluating Language Models for State Tracking

Shubham Toshniwal
Research Scientist, FAIR NYC
Entity Tracking in Chess

Test out ideas for entity tracking via language models in chess

Why Chess? *Simple, closed domain*

Entities: Chess pieces
Entity State: Piece Location
Learning Chess Blindfolded

Transformer Language Model

e2e4
Learning Chess Blindfolded

g1f3  d7d5  g2g3

d2d4  d7d5  g1f3

e2e4  e7e5  g1f3
Algebraic Notation

Position Naming

- Piece Types: Rook (R), Knight (N), Bishop (B), Queen (Q), King (K), Pawn (P)

- Diagram showing the algebraic notation and position naming on a chessboard.
Algebraic Notation

Position Naming

Piece Types

- Rook: R
- Knight: N
- Bishop: B
- Queen: Q
- King: K
- Pawn: P
Chess Notation

Translation of moves
Chess Notation

Translation of moves

e2e4  (Pawn) moved from e2 to e4
Chess Notation

Translation of moves

<table>
<thead>
<tr>
<th>Move</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e2e4</td>
<td>(Pawn) moved from e2 to e4</td>
</tr>
<tr>
<td>e7e5</td>
<td>(Pawn) moved from e7 to e5</td>
</tr>
</tbody>
</table>
Chess Notation

Translation of moves

<table>
<thead>
<tr>
<th>Move</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e2e4</td>
<td>(Pawn) moved from e2 to e4</td>
</tr>
<tr>
<td>e7e5</td>
<td>(Pawn) moved from e7 to e5</td>
</tr>
<tr>
<td>g1f3</td>
<td>(Knight) moved from g1 to f3</td>
</tr>
</tbody>
</table>
Chess Notation

Translation of moves

<table>
<thead>
<tr>
<th>Move</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e2e4</td>
<td>(Pawn) moved from e2 to e4</td>
</tr>
<tr>
<td>e7e5</td>
<td>(Pawn) moved from e7 to e5</td>
</tr>
<tr>
<td>g1f3</td>
<td>(Knight) moved from g1 to f3</td>
</tr>
</tbody>
</table>

...
Entity Tracking Task: Ending Square

Chess Notation allows for probing for entity state via prompting!

e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 f1??
Entity Tracking Task: Ending Square

Chess Notation allows for probing for entity state via prompting!

e2e4  e7e5  g1f3  b8c6  d2d4  h7h6  f1g1
Entity Tracking Task: Ending Square

Chess Notation allows for probing for entity state via prompting!

e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 f1g2
Entity Tracking Task: Ending Square

Chess Notation allows for probing for entity state via prompting!

```
e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 f1b5
```
Randomly Annotated Piece Type (RAP)

Can a language model benefit from the knowledge of piece types?
Randomly Annotated Piece Type (RAP)

Can a language model benefit from the knowledge of piece types?

Randomly introduce piece types in text sequences during training

Vanilla Training   e2e4 e7e5 g1f3 b8c6 d2d4 h7h6
Randomly Annotated Piece Type (RAP)

Can a language model benefit from the knowledge of piece types?

Randomly introduce piece types in text sequences during training

| Vanilla Training                               | e2e4  e7e5  g1f3  b8c6  d2d4  h7h6 |
| + RAP (p=15)                                   | e2e4  e7e5  N   g1f3  b8c6  d2d4  h7h6 |

---

<table>
<thead>
<tr>
<th>Piece Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knight</td>
</tr>
<tr>
<td>Pawn</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>P</td>
</tr>
</tbody>
</table>
Randomly Annotated Piece Type (RAP)

Can a language model benefit from the knowledge of piece types?

Randomly introduce piece types in text sequences during training

<table>
<thead>
<tr>
<th>Training Type</th>
<th>Move Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla Training</td>
<td>e2e4 e7e5 g1f3 b8c6 d2d4 h7h6</td>
</tr>
<tr>
<td>+ RAP (p=15)</td>
<td>e2e4 e7e5 N_g1f3 b8c6 d2d4 h7h6</td>
</tr>
<tr>
<td>+ RAP (p=50)</td>
<td>P_e2e4 e7e5 N_g1f3 b8c6 d2d4 P_h7h6</td>
</tr>
<tr>
<td>+ RAP (p=100)</td>
<td>P_e2e4 P_e7e5 N_g1f3 N_b8c6 P_d2d4 P_h7h6</td>
</tr>
</tbody>
</table>
Randomly Annotated Piece Type (RAP)

Can a language model benefit from the knowledge of piece types?

Randomly introduce piece types in text sequences during training

<table>
<thead>
<tr>
<th>Vanilla Training</th>
<th>e2e4 e7e5 e1f3 b8c6 d2d4 h7h6</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ RAP (p=15)</td>
<td>e2e4 e7e5 Ng1f3 b8c6 d2d4 h7h6</td>
</tr>
<tr>
<td>+ RAP (p=50)</td>
<td>P e2e4 e7e5 Ng1f3 b8c6 d2d4 Ph7h6</td>
</tr>
<tr>
<td>+ RAP (p=100)</td>
<td>P e2e4 P e7e5 Ng1f3 Nb8c6 P d2d4 Ph7h6</td>
</tr>
</tbody>
</table>

Inference: e2e4 e7e5 g1f3 b8c6 d2d4 h7h6
Entity Tracking Task: Starting Square

Training with RAP also allows for directly probing for piece location

e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 B??
Entity Tracking Task: Starting Square

Training with RAP also allows for directly probing for piece location

e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 Bf1
Entity Tracking Task: Starting Square

Training with RAP also allows for directly probing for piece location

e2e4 e7e5 g1f3 b8c6 d2d4 h7h6 Bc1
Entity Tracking Results

![Graph showing GPT-2 performance across different dataset sizes](image)

- **Train-S**: 60% accuracy
- **Train-M**: 90% accuracy
- **Train-L**: 100% accuracy
Entity Tracking Results

![Graph showing comparison of LgM Accuracy between GPT-2 and GPT-2 + RAP across different dataset sizes (Train-S, Train-M, Train-L). The graph illustrates the performance increase with the addition of RAP for all dataset sizes.]
Error Categories

Automated error analysis possible for domains such as chess

Error categories:
  Syntax
  Path Obstruction
  Pseudo Legal
Error Category: Syntax

Queen trying to move like a knight
Error Category: Path Obstruction

Bishop eager to retreat
Protect the king first
Language Modeling Results

![Graph showing the relationship between perplexity and RAP probability for different training set sizes: Train-S, Train-M, and Train-L. The graph illustrates how perplexity decreases as RAP probability increases for each training set size.](image)
Takeaways

Proposed chess as a testbed for entity tracking in language models

Data augmentation using RAP improves both entity tracking and language modeling results for low data settings