Learning a Game Commentary Generator with Grounded Move Expressions

Hirotaka Kameko
Graduate School of Engineering,
The University of Tokyo
Bunkyo, Tokyo, Japan
kameko@logos.t.u-tokyo.ac.jp

Shinsuke Mori
Academic Center for Computing and Media Studies,
Kyoto University
Sakyo, Kyoto, Japan
forest@i.kyoto-u.ac.jp

Yoshimasa Tsuruoka
Graduate School of Engineering,
The University of Tokyo
Bunkyo, Tokyo, Japan
tsuruoka@logos.t.u-tokyo.ac.jp

***Abstract*—This paper describes a machine learning-based approach for generating natural language comments on Shogi games. We generate comments by using a discriminative language model trained with a large amount of Shogi game records and comments made by human experts. Central to our method is accurate mapping of move expressions appearing in experts' comments to game states (*i.e.* positions) of Shogi, because the discriminative language model is trained with textual expressions paired with corresponding Shogi positions. We describe how such mapping can be performed by using evaluation information obtained from a Shogi program. Experimental results show that we can actually generate helpful comments for some positions.**

I. Introduction

Natural language is arguably one of the most important communication media that machines could use when they present information to humans, but it has been largely underexploited in current AI systems mainly due to the difficulties of natural language generation problems. In this paper, we focus on a particular subproblem of natural language generation and aim to develop a computer system that can analyze a given game record of Shogi (Japanese chess) and express its "thought" in natural language.

Today, computer Shogi programs are as strong as human professional players, and many of the strong programs are available as commercial or freeware applications. It is now common for Shogi fans to use a Shogi program when they watch games played by professional Shogi players, because the evaluation information provided by the program often helps them a lot to understand the implication of the moves played in the game in real time. The problem is, however, that the current programs can only provide evaluation information in a very crude form (*e.g.* a sequence of moves and an evaluation score) — no program can, for example, explain why a certain move is better than others. Thus, people still need a high level of Shogi expertise to fully appreciate the game even with the help of Shogi programs.

A different (and perhaps traditionally more common) source of evaluation information available to people for understanding games is the natural language comments made by experts. Most high-profile Shogi matches are broadcast with such comments and they are indispensable for novice players to enjoy watching the games. Unfortunately, however, such comments are very costly to produce in terms of human labor, and are thus not available in all games played by professional Shogi players, let alone thousands of games played every day by amateur players.

In this paper, we aim to build a computer system that can automatically generate a commentary for a given Shogi game. Our method consists of two machine learning steps. First, we predict words that characterize input positions, which should appear in comments. Then, we generate comments with these words and a log-linear language model. This model is trained with actual comments made by human experts.

Central to our method is accurate mapping of move expressions appearing in experts' comments to game states (*i.e.* positions) of Shogi, since the discriminative language model is trained with textual expressions paired with corresponding Shogi positions. We describe how such mapping can be performed by using evaluation information obtained from a Shogi program.

II. Related Work

In this section, we describe recent work on natural language generation for games and other applications.

A. Commentary Generation for Games

Sadikov et al. [1] describe a rule-based approach for generating comments on a chess game. Their system generates a comment with hand-crafted rules according to the values of certain features of the evaluation function. Since the evaluation function of a chess program is designed manually, the generation model uses these useful and intuitively clear features. By contrast, most strong computer Shogi programs use an evaluation function built by machine learning with combinational features of pieces [2], which makes it difficult to understand the meanings of feature values.

Kaneko [3] proposed a template-based method for generating comments on a Shogi game. Their system presents information about the result of a game tree search (*e.g.* principal variations and evaluation values) using predefined generation templates. The system provides not only candidate moves and evaluations, but also various search results such as checkmates and threatmates, and null move search. These kinds of information are helpful in watching games, but the variety of the comments generated by the system is much smaller than that of human experts' comments. This is a fundamental limitation of a template-based approach.

A template-based approach such as Kaneko [3] can generate useful and grammatical sentences for certain situations. Our machine learning-based approach complements template-based methods in that it allows us to generate a variety of comments that are hard to generate with manually designed templates.

B. Natural Language Generation for Other Applications

Ratnaparkhi [4] proposed a generation method using a language model. A language model assigns generation probabilities to sequences of words. For example, with an n-gram language model, a generation probability is defined as the product of the generation probability of each word defined with the previous $n-1$ words. When we generate a sentence, we perform a search to find the sentence which maximizes the generation probability.

Kiros et al. [5] proposed natural language models that can be conditioned on other modalities. They proposed multimodal log-bilinear models, which are extensions of a log-bilinear model [6] with other modalities. They combine feature vectors of linguistic and nonlinguistic context feature vectors. They showed that the nonlinguistic context feature improves the performance of a language model.

Dani et al. [7] proposed a log-linear language model that captures temporal dynamics. They define a deviation of word frequency derived from nonlinguistic context features. They showed that their method improves in economy-related text.

C. Language Model-Based Natural Language Generation

A traditional approach for natural language generation is that of using a language model. A language model assigns a probability to a sequence of words. For example, with an n-gram-based language model, the probability of a sentence $S = w_1, w_2, \ldots, w_M$ is defined as

$$P(S) = \prod_{i=1}^{M} P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \simeq \prod_{i=1}^{M} P(w_i \mid w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}) \qquad (1)$$

$$P(w_i \mid w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}) = \frac{Count(w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}, w_i)}{Count(w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1})}, \qquad (2)$$

where M is the length of the sentence and $Count(Sequence)$ is the number of occurrences of $Sequence$ in the corpus. When generating a sentence, we output the S which maximizes $P(S)$.

III. Shogi and Commentary

Shogi is a chess-like board game and is also known as Japanese chess. Figure 1 shows its starting setup. In addition to six kinds of chess-like pieces, Shogi has three kinds of pieces: gold (at 4a, 4i, 6a, and 6i), silver (3a, 3i, 7a, and 7i), and lance (1a, 1i, 9a, and 9i). The rule of Shogi is different from that of chess in two respects. One is that the player can capture the opponent's pieces and drop a captured piece in a subsequent turn. This rule makes Shogi more complicated, especially in end-game positions. The other is that almost all pieces, except golds and kings, can promote if they move to, or move from the seventh, eighth, or ninth rank. In chess, only a pawn can promote if it proceeds to the eighth rank, and thus promotion is infrequent. In Shogi, however, promotion happens frequently.

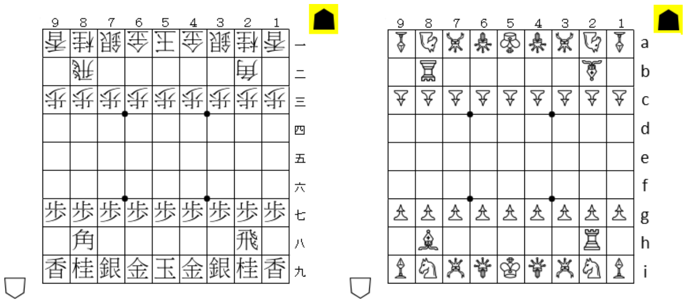


Fig. 1. Starting setup of Shogi (right: expression like chess)

In Shogi, comments about moves and positions (*i.e.* game states) usually consist of textual descriptions in natural language and *move expressions*. A move expression consists of a turn expression (▲ for Black and △ for White[1]), a position expression, a piece expression, and other information such as dropping, promoting, and disambiguation. In this work, we need to know which positions the move expressions actually correspond to. For example, the comment "△１四香とすれば決戦" (If White plays Lx1d, then the game phase shifts to the end game) contains a move expression "△１四香" (White + Lx1d). Comments sometimes contain a long sequence of moves. "▲４六角△同成桂▲同飛は△３七馬が両取りになって失敗する。" (If they play B-4f +Nx4f Rx4f, White will play +B-3g and it forks a rook and a knight) contains a sequence of three moves and a move following them. "+B-3g forks" is a comment about +B-3g. However, this move is not a move from the current position, but from the position realized by playing the sequence of the moves. Thus we need to figure out which position the comments are about and which position the move expression in the comment is from.

IV. Game Commentary Generator

Our generation method consists of two machine learning steps. First, we predict the words that characterize the input position which may appear in the comments. Then, we generate comments containing some of the characteristic words using a language model. Figure 2 shows the overview of our system. This two-step method facilitates analyzing the error. For example, we can easily know which step causes the error.

A. Grounding Comments to State Spaces

Before training a language model for comment generation, we need to associate comments with positions because many comments are not about the current positions. In the field of natural language processing, this kind of mapping between text and an external world is often called *grounding* [8]. We perform this grounding by a rule-based approach using evaluation values of a computer Shogi program. In this section, we introduce *a commented tree* and *candidate trees* and describe how to generate them.

[1]In Shogi, Black plays first (in contrast, White plays first in chess).

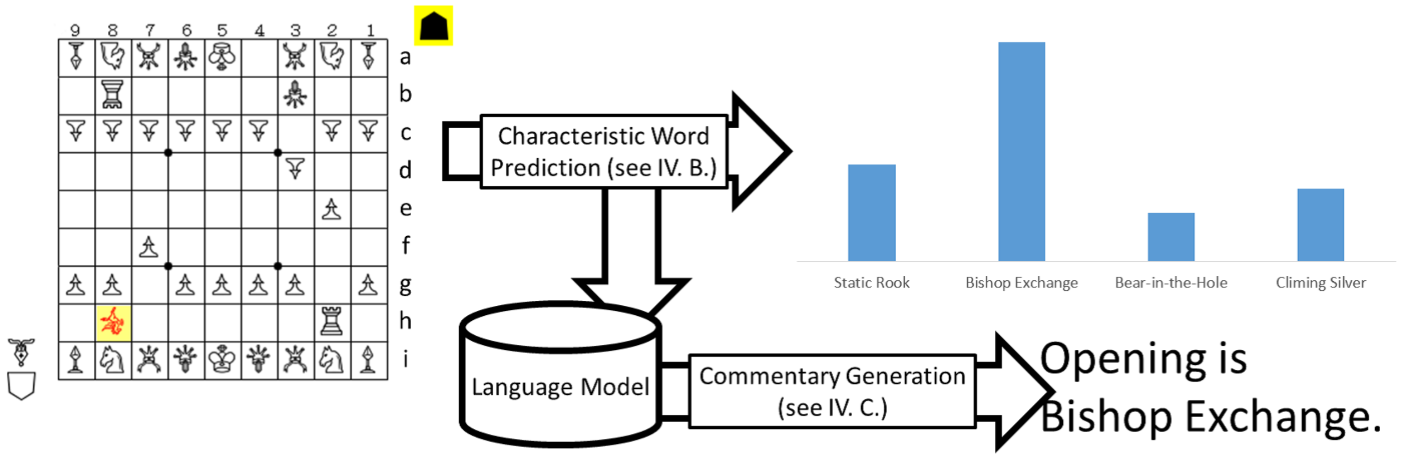

9 8 7 6 5 4 3 2 1
a b c d e f g h i
Characteristic Word Prediction (see IV. B.)
Static Rook
Bishop Exchange
Bear-in-the-Hole
Climing Silver
Language Model
Commentary Generation (see IV. C.)
Opening is Bishop Exchange.

Fig. 2. Overview of our System

Let's say, for example, a human expert makes a comment "△２五飛と回られては▲２八歩と使わされてしまう" (If White plays **R-2e**, Black will have to play **P*2h**) for the right position of Fig. 3. It means that if Black had not played the last move (R-2f), Black would have to drop a captured pawn (so Black should play R-2f). On the other hand, White can actually play R-2e in *this* position as well and Black can also play P*2h after the move. However, these two moves are quite bad and senseless (only to give a rook to the opponent player).

In Fig. 3, the two trees below the position are legal subtrees and the boxed ones are the correct trees which the comments refer to. We define the correct tree as a commented tree and the set of legal subtrees as candidate trees.

We propose a two-step method to generate a commented tree. First, we enumerate candidate trees by a rule-based algorithm. Then, we choose a commented tree using evaluation values of a Shogi program.

We use the following rules for enumerating candidate trees.

1) Initialize a tree with the current position and the previous n moves (in this paper $n = 3$).
2) Extract move expressions by using regular expressions and split the comment, *e.g.*, ここで / △７六飛 / のマネ将棋は / ▲２二角成 / に取る駒がなく、終わってしまう。
3) Add "Pass" to the legal moves[2].
4) If a legal move is found, add the move with the rules below if the move has not been added.
    - If the previous phrase is a move expression, add it below.
    - If the previous phrase is one of some symbols like "$\sim$", add Pass below and add the move below the Pass (for example, White's move $\sim$ White's move $\sim$ White's move).
    - If the previous phrase contains a symbol such as "(1)" "(2)", "(a)" "(b)", add it to the same parent node.
    - If the previous phrase does not contain "。" (punctuation mark in Japanese), add it under the previous move expression.

[2]In actual games Pass moves are illegal, but they are often needed in commentary.

Next, we choose the commented tree from the candidate trees. In this work, we rely on the observation that expert players sometimes make a comment about bad moves for the purpose of explanation, but most of commented moves are good moves. When they mention a bad move, they usually explain why it is a bad move by showing the right move, rather than mentioning another bad move. Therefore, we use evaluation values of a Shogi program to choose a tree which does not contain a sequence of bad moves. We assume that bad moves do not appear more than once in the same tree.

We define a tree score function $Ev_T(t)$ as below.

$$Ev_M(m) = |Ev_P(pos_p, d) - Ev_P(pos_c, d-1)|$$
$$Ev_T(t) = \sum_{m \in t} Ev_M(m) - \max_{m \in t}(Ev_M(m)) + B_P + B_B,$$

where $Ev_P(pos, d)$ is the evaluation value that a Shogi program outputs as the result of a game-tree search for a position pos (search depth= d); m and t are a move and a candidate tree, respectively. pos_p is the position before playing the move m and pos_c is the position after playing m. Note that $Ev_M(m)$ represents how bad m is, and worse move m has larger $Ev_M(m)$. If m is the move which the computer program returns, $Ev_P(pos_p, d)$ and $Ev_P(pos_c, d-1)$ are equal, but if m is a bad move, the evaluation value drops sharply. B_P is a bias for Pass moves and B_B is a bias for branching. In Shogi, an evaluation value fluctuates intensely in end-game positions [9], so we change these biases according to how hot the position is. First, we calculate a phase value (0–127) by using the information on positions, captured pieces, and promoted pieces. This value is low in opening positions and high with end-game positions. In this work, we use

$$B_B = 100 + 400 \times PHASE/127 \tag{3}$$
$$B_P = B_B \times 3. \tag{4}$$

$Ev_T(t)$ shows how bad moves in the tree t are, so we choose the tree with the minimum score as the commented tree. After we get the commented tree, we use pairs of moves and split

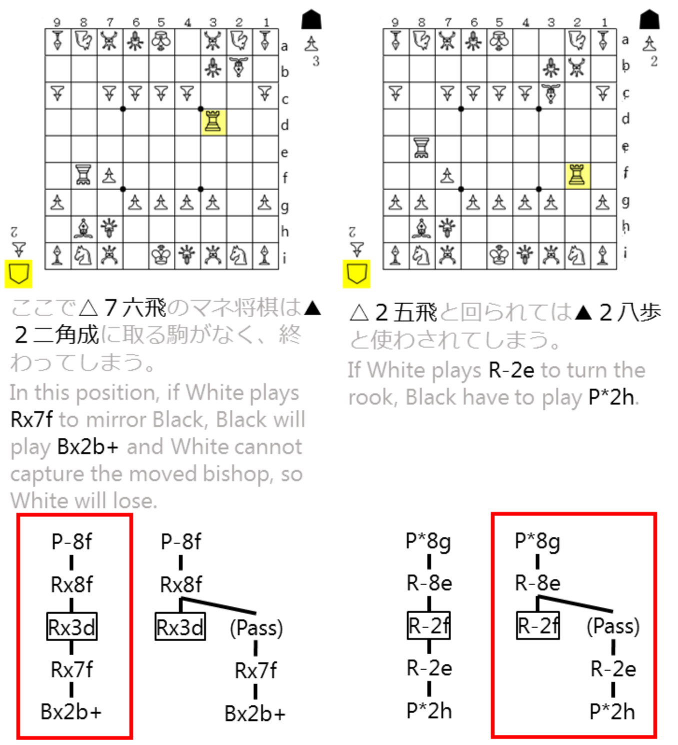

ここで△７六飛のマネ将棋は▲２二角成に取る駒がなく、終わってしまう。
In this position, if White plays Rx7f to mirror Black, Black will play Bx2b+ and White cannot capture the moved bishop, so White will lose.
△２五飛と回られては▲２八歩と使わされてしまう。
If White plays R-2e to turn the rook, Black have to play P*2h.
P-8f
Rx8f
Rx3d
Rx7f
Bx2b+
P-8f
Rx8f
Rx3d
(Pass)
Rx7f
Bx2b+
P*8g
R-8e
R-2f
R-2e
P*2h
P*8g
R-8e
R-2f
(Pass)
R-2e
P*2h

Fig. 3. Examples of "commented trees" and "candidate trees"

comments after the move expressions as the training examples for the language model.

In this paper, we use Gekisashi [10] as the Shogi program that performs game-tree search and returns evaluation values. Note that, however, our method only requires the result of evaluation, so it can be used with any Shogi programs as long as they return an evaluation value.

B. Prediction of Characteristic Word

We also build a machine learning model which predicts words that characterize a given position.

For the prediction of such words, we use the features used in the evaluation function of Gekisashi. Those features include values of pieces, positions of pieces, and piece-piece relations. Computer Shogi programs have strengthen using machine learning with these features. Thus these features are very important for understanding positions, and we also expect that these are good at depicting positions. For example, if Black's gold and silver are near to Black's king, the gold and silver are useful in protecting Black's king (hence they have high weights in the evaluation function), and if White's pieces are near to Black's king, they are considered to be threatening Black's king. We also use move-piece relations as features. For example, if a piece which is threatened by the opponent's piece, the aim might be dodging.

The prediction model outputs a d-dimensional real-valued vector, each of whose elements indicates whether a particular word should appear in a comment or not, where d is the size of vocabulary. We use a three-layer perceptron with binary outputs for building the model.

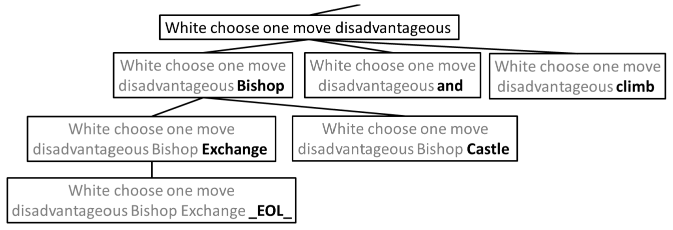

White choose one move disadvantageous
White choose one move disadvantageous Bishop
White choose one move disadvantageous and
White choose one move disadvantageous climb
White choose one move disadvantageous Bishop Exchange
White choose one move disadvantageous Bishop Castle
White choose one move disadvantageous Bishop Exchange _EOL_

Fig. 4. Generating a sentence with best first search

C. Commentary Generation

This section describes a method which automatically generates a comment by using a language model and characteristic words given by the model described in the previous section. Given a position p, we define a generation probability of sentence $S = w_1, w_2, \ldots, w_n$ as

$$\begin{aligned}
&P(S \mid p) \\
&= P(S_N \mid length(S_N) = n) \\
&\times P(w_1 \mid p) \times P(w_2 \mid p, w_1) \times P(w_3 \mid p, w_1, w_2) \\
&\times \cdots \times P(w_n \mid p, w_1, w_2, \ldots, w_{n-1}) \\
&= P(S_N \mid length(S_N) = n) \\
&\times \prod_i^n P(w_i \mid p, w_1, w_2, \ldots, w_{i-1}). \qquad (5)
\end{aligned}$$

In this formula, $P(S_N \mid length(S_N) = n)$ is the probability that a sentence with n words will be generated. We search for S which maximizes $P(S \mid p)$.

In this work, we estimate the generation probability of each word by using a log-linear model (also known as a softmax regression or a maximum entropy model). We define the generation probability of words as

$$P(w_i \mid p, w_1, \ldots, w_{i-1}) = \frac{\exp(W_{w_i}^T \phi(p, w_1, \ldots, w_{i-1}))}{\sum_j \exp(W_{w_j}^T \phi(p, w_1, \ldots, w_{i-1}))}, \qquad (6)$$

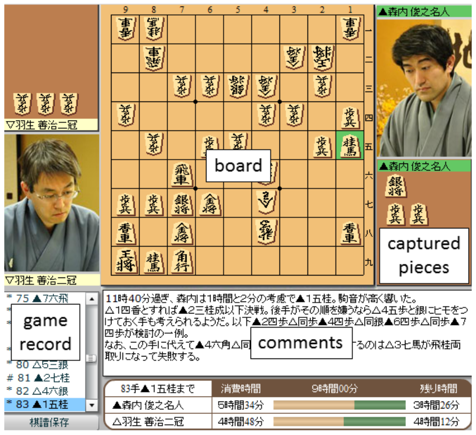
where $\phi(p, w_1, \ldots, w_{i-1})$ is a feature vector and W_{w_i} is the weight vector for w_i. We use the predicted characteristic words as context features, and the previous two words and bag of words as linguistic features. We search for S which maximizes the generation probability by a best-first search method depicted in Fig. 4. Since the time complexity and space complexity become prohibitive if we search the whole tree, we employ beam search. Each node has string $S_{imp} = w_1, w_2, \ldots, w_k$ and the generation probability $P(S_{imp} \mid p) = \prod_i^k P(w_i \mid p, w_1, w_2, \ldots, w_{i-1})$. Until the best node reaches a terminal symbol, we expand the node with all of words in a vocabulary. If an expanded node has a lower probability than other leaf nodes, we discard it. If the last word is a terminal symbol, calculate $P(S \mid p)$ by (5), and if $P(S \mid p)$ is higher than all of P_{inf}, the system outputs S.

V. Dataset and Preprocessing

We used game records of Shogi commented by human experts. We used game records of Meijin title matches (the most high-profile matches in professional Shogi) and preliminary tournaments (Jun'i-sen in Japanese). The records are

```
(header)
手数----指手-- # game record starts from this line
* a line starts with "*" is a comment line
   1  ７六歩(77)    ( 0:00/00:00:00)
* The upper line is a move expression (1. 7f from 7g)
* comments about the position after 7f
   2  ３四歩(33)    ( 0:00/00:00:00)
                    ⋮
```

Fig. 5. KIF format



* 75 ▲7六飛
* 80 △5三銀
# 81 ▲2七桂
* 82 △4六銀
* 83 ▲1五桂
棋譜保存

11時40分過ぎ、森内は1時間と2分の考慮で▲1五桂。駒音が高く響いた。
△1四香とすれば▲2三桂成以下決戦。後手がその順を嫌うなら△4五歩と銀にヒモをつけておく手も考えられるようだ。以下▲2四歩△同歩▲4四歩△同銀▲6四歩△同歩▲7四歩が検討の一例。
なお、この手に代えて▲4六角△同 るのは△3七馬が飛桂両取りになって失敗する。

83手▲1五桂まで	消費時間	9時間00分	残り時間
▲森内 俊之名人	5時間34分		3時間26分
△羽生 善治二冠	4時間48分		4時間12分

Fig. 6. Example of Game and Comment Viewer

distributed[3], in KIF format like Fig. 5. Figure 6 is a snapshot of a match viewer, showing a position and the comments being made by experts for the position.

Table I shows the number of comments made by human experts and the number of games. A, B1, B2, C1, and C2 are the classes of preliminary tournaments. Class-A tournament is the highest one, and the winner of the class-A tournament challenges the Meijin title holder. Meijin title matches and games of class-A tournament have a higher profile, having more comments per one game than the games in the lower classes.

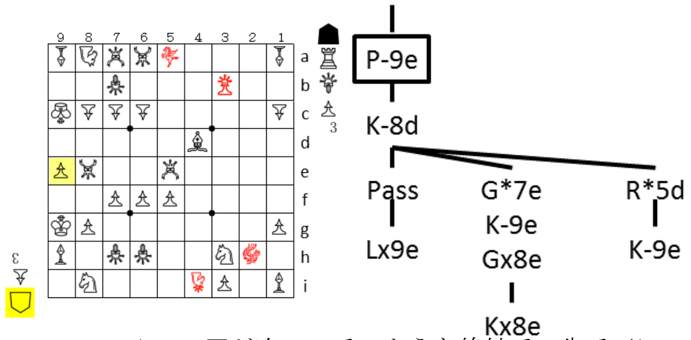
The comments are written in Japanese. Since it has no whitespace between words, we segment sentences into words using an open-source word segmenter, KyTea [11].

These comments sometimes contain information that is not directly relevant to the game such as what the players ate, how they look and how long they thought. These are interesting pieces of information for human readers, but our models should not use these comments for training because they cannot be associated to a state of the game and thus work as noise when

TABLE I. RELATIONSHIP BETWEEN CLASS AND THE AMOUNT OF COMMENTS (NUMBER OF COMMENTS / NUMBER OF GAMES)

Year	Meijin	A	B1	B2	C1	C2
70th	1,979/7	8,259/45	6,816/78	8,235/120	11,363/164	13,323/217
69th	1,185/4	8,124/45	6,392/78	6,484/120	8,407/157	10,218/213
68th	1,971/7	8,213/45	6,927/78	6,801/120	7,622/155	8,615/217
67th	1,234/6	7,126/45	4,898/78	5,534/117	7,156/155	8,013/215
66th	1,382/7	5,359/45	4,513/78	4,422/109	5,606/144	7,548/225
65th	728/6	4,388/46	2,875/78	3,373/115	4,461/140	5,848/227
64th	0/7	720/45	372/78	425/115	693/148	1,007/228

[3]http://www.meijinsen.jp/ (in Japanese)



Comments : △８四玉が次の一手のような絶妙手。先手玉には△９五香の詰めろがかかり、後手玉には▲７五金△９五玉▲８五金と追われても、△同玉が逆王手となる。また▲５四飛の王手も△９五玉と逃げて後手の勝ち。(略)
(K-8d is the best move. Black is in threatmate from Lx9e and if Black plays G*7e K-9e Gx8e, White's Kx8e is a mate move. If Black plays R*5d, White Plays K-9e and White will win.)

Fig. 7. Example of Grounding

training our model. In this paper, we use only comments which are associated to a state of the game for training.

VI. EXPERIMENTS AND RESULTS

A. Adapting a Word Segmenter

Before we process the text, we adapted KyTea to the Shogi domain. We annotated 469 sentences on Shogi with word boundary information. To evaluate the accuracy of the analyzer, we use 299 sentences as the training set and 170 sentences as the test set. Table II shows the results. The F-score of the analyzer for general documents is about 0.97 [11], so the performance of the default model on text in the Shogi domain is bad. The results also show that our domain adaptation has a good effect on performance.

B. Grounding Comments to State Spaces

We generated commented trees for the positions with comments including move expressions using the proposed method. Figure 7 shows a position where the proposed method generated the correct commented tree. These comments contain eight move expressions, and seem quite complicated. However, this position includes many mate moves, so there are few legal moves. In addition, the position is about checkmate, so the commented moves are valid moves.

We generated candidate trees by using the methods described in section IV. A. The comments including "N 手目" (N-th move) often refer to positions that are very far from the current position, so we ignored these comments in this experiment. If the number of candidate trees in one position

TABLE II. RESULT OF WORD SEGMENTATION

Model	Precision	Recall	F-score
Default	91.00	92.20	91.60
Adapted	96.71	96.40	96.56

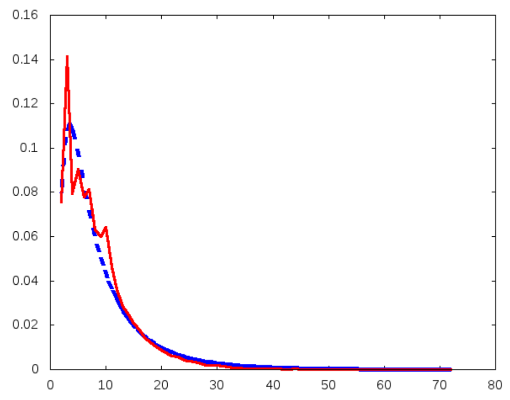


Fig. 8. Distribution of length of comments (solid red: real distribution, dotted blue: approximated distribution

exceeded 50, we stopped generating and treated this case as failed.

We applied the proposed method to 54,084 positions and obtained at least one candidate tree for 44,166 positions (81.2%). The generation process was stopped in $4,560/9,918$ positions because of the number of candidate trees. We manually checked 50 positions in which our proposed method could not generate a candidate tree.

Table III shows the result. We can observe that the errors in text are the most popular. Because the comments we used in this experiment are flash reports written by human, they include many mistakes. The result shows that our method can detect some errors.

We also observe many errors caused by move expressions in natural language. For example, Shogi players often use the expression “銀を上がる” (bring up silver). Understanding this kind of natural language expressions is not straight-forward, and we leave it for future work.

We checked 100 positions in which our method generated at least one candidate tree, and found that our method selected the correct commented tree for 79 positions.

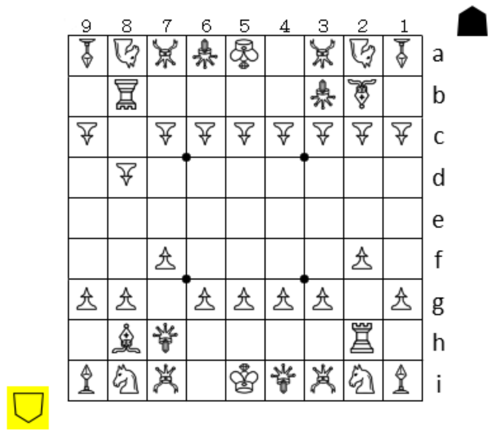
C. Generating Comments

Figure 8 shows the distribution of the lengths of comments. We estimated it with an inverse Gaussian distribution,

$$\sqrt{\frac{\lambda}{2\pi n^3}} \exp\left(\frac{-\lambda(n-\mu)^2}{2\mu^2 n}\right), \tag{7}$$

TABLE III. ERROR ANALYSIS OF GENERATING CANDIDATE TREES

Error type	# of positions
error of text	19
far past move	7
expressions by natural langauge	6
other captured piece	5
about position	5
illegal move	4
leaving mate	3
other game	1



Move: P-8e

Word	Value	Word	Value
角 (bishop)	0.56	手 (hand)	0.35
換わ (exchange)	0.32	し (do)	0.30
損 (loss)	0.29		

Generated: △８五歩と突けば、後手の角換わりになりそうだ (If White plays P-8e, White will choose bishop exchange strategy.)

Move: P-3d

Word	Value	Word	Value
角 (bishop)	0.85	手 (hand)	0.81
横歩 (side pawn)	0.77	取り (capture)	0.71

Generated: △３四歩なら角換わりになり、横歩取りの将棋となる (If White plays P-3d, the game will be bishop exchange strategy, and the game will be side pawn capturing strategy.)

Fig. 9. Position of “Bishop Exchange”

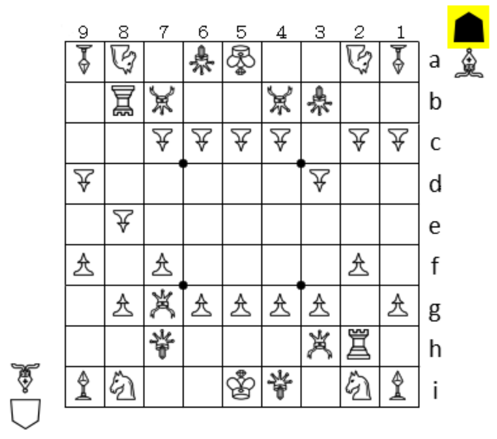
and use it as $P(len(S))$ in equation (5). The estimated parameters are $\lambda = 11.94, \mu = 9.03$. In addition, we restricted the length of a generated sentence to be over 15 words.

Some Shogi comments have little information. For example, if *ME* is a move expression, the comment “*ME* だ。” (played *ME*.) is very short and it does not explain the move. Because most of such comments are short, we use comments over 10 words as the training corpus in this experiment.

A comment can either follow or precede a move expression. For example, a comment “*ME* はよい手だ” (*ME* is a good move) describes a left side move expression, and a comment “囲いを作る *ME*” (*ME* which makes a castle) describes a right side move expression. However, it is very hard to decide which side does the comments describe. In this experiment, we consider that all of the comments describe left side move expressions.

Here we show some examples of the generated comments.

Figure 9 shows a correct example which our method generated. This position is an intermediate position of an opening game called “角換わり戦法” (bishop exchange strategy). Before this move, White can play P-3d instead of this move. If White plays P-3d, the game will go for another opening game called “横歩取り戦法” (side pawn capturing strategy). The

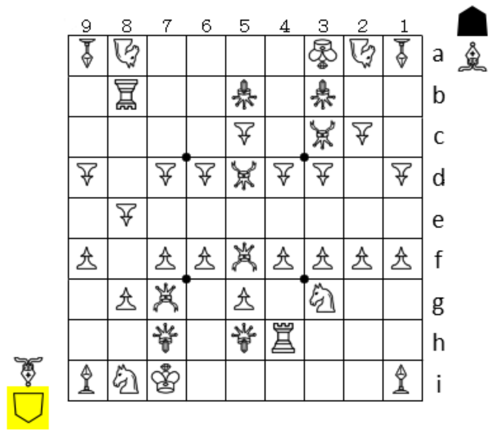


Move: P-4f

Word	Value	Word	Value
腰掛け (reclining)	0.99	指 (action)	0.99
方 (plan)	0.99	銀 (silver)	0.98

Generated: ▲４六歩から腰掛け銀を目指す指し方もあるところだが、 (Black can play P-4f and aim for reclining-silver strategy, but. . .)

Fig. 10. Position of “Reclining Silver”



Move: G5b-4b

Word	Value	Word	Value
玉 (king)	0.47	固め (stabilize)	0.38
あ (be)	0.36	後手 (White)	0.33

Generated: △４二金右と固めるつもりだろう、と言われている (Someone says “White will play G5b-4b to stabilize the castle”)

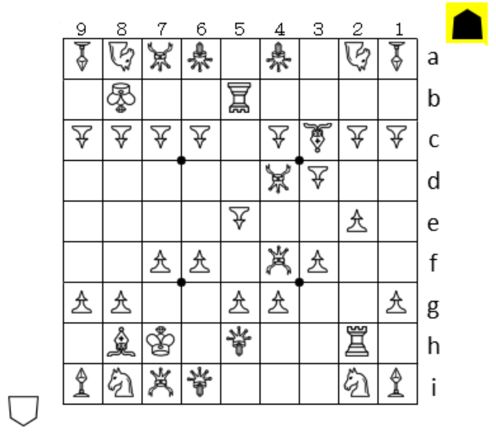
Fig. 11. Position of “Stabilizing the Castle”

opening strategy is fixed by P-8e. Thus the comment about this position and move should be about these strategies. Our system generated a comment about the bishop exchange strategy, and we consider the comment to be correct and accurate.

On the other hand, if we give the move P-3d for the position of Fig. 9, our system generates another comment. In this position, if White plays P-3d, the game will follow either the bishop exchange strategy or the side pawn capturing strategy, and the strategy is not fixed yet. The predicted characteristic words show that the comment should be about both the bishop exchange strategy and the side pawn capturing strategy. Our system generated a comment about these strategies. However, White needs to choose either of those strategies and they cannot happen in the same game, so the generated comment is not correct.

Figure 10 shows a position that occurred after that of Fig. 9. If Black’s silver goes to 5f and Black’s pawn is on 5g (White’s silver goes to 5d and the pawn is on 5c), the strategy is called “腰掛け銀” (reclining silver strategy). This naming comes from the fact that the silver on 5f looks like sitting on the pawn on 5g. In the position shown in Fig. 10, the silver has not reached 5f yet and the silver went to 5f after 14 moves. The generated comment shows one of the viewpoints on this position. The comment ends with “だが、” (, but. . .). This is due to a characteristic of the training data. In this experiment, we did not distinguish the comments about potential positions from the ones that actually happened. For example, in a comment “If Black played move M_1, the game would be C_1, but Black has actually played move M_2 so the game will be C_2”, we treat those two statements equally in the training of the models. This is a source of many problems, including the disagreement of tense.

Figure 11 shows a position where White is going to stabilize its castle. A castle is a defensive position built near



Move: G-6g

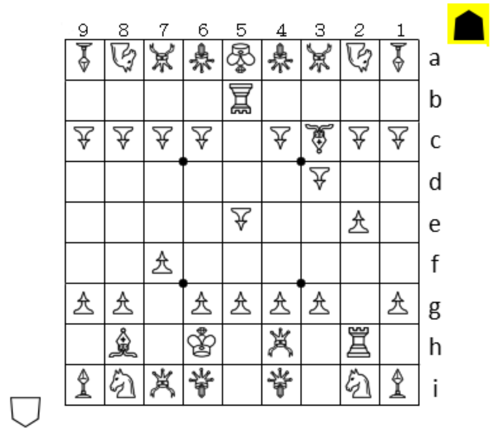
Word	Value	Word	Value
筋 (file)	0.98	手 (hand)	0.86
5	0.50	受け (defend)	0.47

Generated: ▲６七金と上がり、５筋の筋を受ける手がある (Black can play G-6g to defend the fifth file)

Fig. 12. Position of “Defending Fifth File”

the king. In this position, the gold on 3b, the silver on 3c, the knight on 2a, the lance on 1a and the pawns constitute a castle. White played G5b-4b to attach another gold to the castle and stabilize it. This move or position has no proper name like the bishop exchange strategy. The generated comment, “stabilize the castle,” is a general language expression and this shows that our system can make useful comments without using the names of the strategies.

Figure 12 shows a position where Black can (and probably should) bring up a gold to defend 5f. White’s strategy is to



Move: P-3f

Word	Value	Word	Value
急戦 (quick attack)	0.76	超速 (super quick)	0.52
見せ (hint)	0.28	突 (bring up a pawn)	0.16

Generated: ▲３六歩を見せて急戦を見せて急戦を見せて急戦を見せた (Black hints to play P-3f and hints to attack quickly and hints to attack quickly and hints to attack quickly.[4])

Fig. 13. Position of “Quick Attack”

attack the fifth file using the rook and other pieces, which is called “中飛車” (central rook strategy). In this position, White will attack with a rook, a bishop, a silver, and a pawn. Black needs to defend the file at 5f, but at the moment only a single pawn defends 5f. Thus Black brings up the gold on 5h to add an effect. The move is a defensive move because the opponent player is about to attack. To understand that, we need to count not only Black’s pieces, but also White’s pieces. The generated comment shows that our system can generate sentences using information on the whole board.

Figure 13 shows a position where Black brings up a pawn. This opening is called “超速３七銀” (super quick attack S-3g). This is one of the quick attack strategies. In this position Black played P-3f instead of K-7h to make a castle. Black chose a quick attack strategy when Black played this move. Hence one should make a comment about the quick attack strategy for this position and our system indeed made a comment about that. However, the generated comment is grammatically wrong. The main reason is that our generation model only evaluates local probabilities.

These results show that our proposed system can generate helpful comments about some positions. However, there is much room for improvement.

VII. Conclusion

In this paper, we have proposed a rule-based method for grounding move expressions to the state space of Shogi and a machine learning-based method for generating natural language comments. We trained our model with game records and comments made by human experts. We have shown experimentally that our system can generate helpful comments about some positions.

To further improve our system, we need to improve both grounding and generating methods. For about 20% of the comments with move expressions our system failed to generate candidates, and wrong candidates were chosen for about 20% of the comments with candidate trees. These error rates, especially the latter, directly affect the performance of the trained model. Although it is hard for a program to understand what natural language comments mean, it may be possible to improve the performance by using more elaborate rules.

Our generation model does not use any heuristics of Shogi except for the features of the evaluation function. Thus, our model can be applied to other games or domains. The main problem of generation is the lack of information. To generate precise comments, we need to consider using richer information. Our generation algorithm maximizes the product of the local probabilities. It would be interesting to use a global language model, which can consider what are expressed in different parts of a sentence at the same time.

Acknowledgment

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Number 26540190.

References

[1] Aleksander Sadikov, Martin Možina, Matej Guid, Jana Krivec, and Ivan Bratko. Automated Chess Tutor. In *Proceedings of the 5th International Conference on Computers and Games*, pages 13–25, 2006.

[2] Kunihito Hoki. Optimal control of minimax search results to learn positional evaluation. In *11th Game Programming Workshop*, number 2, pages 78–83, 2006. (in Japanese).

[3] Tomoyuki Kaneko. Evaluation of real-time commentary generated by computer shogi program. *IPSJ Journal*, 53(11):2525–2532, 2012. (in Japanese).

[4] Adwait Ratnaparkhi. Trainable Approaches to Surface Natural Language Generation and Their Application to Conversational Dialog Systems. *Computer Speech & Language*, 16(3-4):435–455, July 2002.

[5] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 595–603, 2014.

[6] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648, 2007.

[7] Dani Yogatama, Chong Wang, Bryan R. Routledge, Noah A. Smith, and Eric P. Xing. Dynamic language models for streaming text. *Transactions of the Association for Computational Linguistics*, 2:181–192, 2014.

[8] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[9] Hirotaka Kameko, Akira Ura, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Improving the performance of probcut by using positional features in shogi. In *17th Game Programming Workshop*, pages 36–43, 2013. (in Japanese).

[10] Yoshimasa Tsuruoka, Daisaku Yokoyama, and Takashi Chikayama. Game-tree search algorithm based on realization probability. *ICGA Journal*, 25(3):145–152, 2002.

[11] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, 2011.

[4]This is a literal translation.