CGDL を用いたトリックテイキングゲーム 自動生成手法の提案と評価

牧野 貴斗 [1]　濱川 礼 [2]
[1] 中京大学大学院 工学研究科　情報工学専攻　[2] 中京大学 工学部

概要

本研究では CGDL(A Card Game Description Language) を用いてトリックテイキングゲームのルールを記述し深層学習で生成を行いトリックテイキングゲームを自動生成する手法を提案する．ゲームの自動生成分野では遺伝的アルゴリズムを用いてチェッカーや Go などのバランスの取れたボードゲームを設計する試み [1] などがあるがこれらの研究では，例えば場のサイズは 3 種類の中から選ぶなど限られたバリエーションでしかルールを表現できず，単純なゲームしか生成できない．より複雑なゲームを生成するために，本研究では深層学習に着目し，トリックテイキングゲームの学習を行い、ルールをより複雑に生成することを試み，生成したゲームの評価を行った．

1　背景と目的

カードゲームは沢山の人に親しまれており，だれもが一度は遊んだことがあるものである．人の顔を見ながらできるカードゲームは，コミュニケーション能力を養い，ゲームに勝つために考える過程で思考力を養うことができると考えられている [2][3]．しかし，同じゲームを遊んでいるとゲームに飽きてしまったり，つまらなくなることでゲームに集中できなくなる．しかし，新しいゲームをやるとしてもカードゲームは値段が高くかさばるものであり，簡単に買えるものではない．[4] ではボードゲームの発展を阻害している要因をアンケート調査しているが，値段の高さは 7 位に入っている．以上の理由から常に新しいゲームを気軽に始めることが難しい．そこで常に新しいゲームを提供することで新鮮な体験を利用者に提供できればこの問題を解決できるのではないかと考え，トリックテイキングゲームのルールを学習させ，ルールをより複雑に生成することを試み，生成したゲームの評価を行った．

2　関連研究・関連システム

ボードゲーム自動生成の研究として，GDL(Game Description Language) 言語を利用して，チェッカーや Go などのバランスの取れたボードゲームの自動設計を試みている．[1]．この研究では，遺伝的アルゴリズムを用いて盤面のパターンや，駒の種類，勝利条件などあらかじめ用意されたパターンの中からゲームのバランスを考慮したゲームの生成を遺伝的アルゴリズムを用いて生成を行っている．

また，カードゲームの自動生成を目指す試みとして，‘A Card Game Description Language’[5] がある．この研究は，GDL をカードゲームに改良を加えた言語 CGDL を開発，カードゲームの自動生成を GA で試みた研究である．カードゲーム用の言語の為，ボードゲームのようにどのような場や場のサイズなどの細かい設定を必要とせず，行う処理に関しても，より限定的な命令を採用している言語について述べている．

3　提案手法

[1][5] では，ゲームの自動生成を遺伝的アルゴリズムで行っているが，例えば，場のサイズは 3 種類の中からどれかを選ぶなど，限られたバリエーションでしかルールを表現できず，単純なゲームしか生成できない．より複雑なゲームを生成するために，文章生成などの分野で使われている深層学習に注目した．日本語などの文章と比較して，CGDL の記述は厳格に単語の順番が決められており，深層学習を使用すればルール文の記述を行えるのではないかと考えた．本研究の提案手法を図 1 に示す．トリックテイキングゲームのルールを CGDL を用いて記述したデータセットを用いて LSTM モデルで学習を行う．学習後は，学習を終えた LSTM モデルに最初の単語を入れそのあとに続く単語の予測を行う．この作業を繰り返しトリックテイキングゲームのルール

の生成を行う．なお，CGDL の文法とトリックテイキングゲームのルールの説明は [7] で述べている．

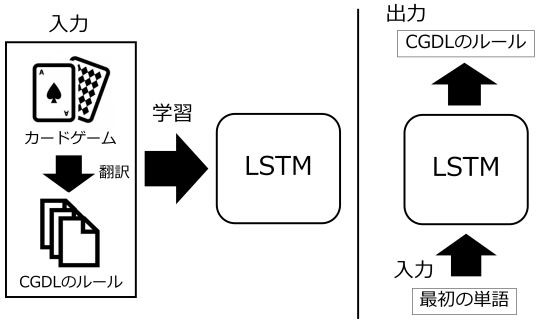

図 1　提案手法

3.1　データセット

今回使用するトリックテイキングゲームはトランプゲーム大全 [8] に記載されているゲームの内 85 個を使用している．

3.2　CGDL の命令の追加

トリックテイキングゲームを表記する際に既存の命令だけでは表現しきれない以下のような状況がある為，表 1 のコマンドを追加する．

- 同じマークのカードを出す
- 既定回数処理を繰り返す
- トリック数を数える
- 得点を失う

行動の種類‘else’は最初に出されたカードのマークと同じマークのカードを出さなくてはならないような状況でマークのカードが出せない時に他のカードを出すという命令を表現するために追加した．アクション‘loop’はトリックテイキングゲームでは決められた回数ループを繰り返す必要があるが，その回数を既存の言語使用では表現できないため追加した．アクション‘trick’は各プレイヤーが取得したトリック数を数えるために追加した．アクション loss は得点を失うという状況を表現するために追加した．

4　システム構成

本手法の構成図を図 2 に示す．LSTM 学習部では，CGDL を用いてルールを記述し，記述したルールを一定のルールに従ってデータセット化し，生成したデータセットを DataAugmentation を行い，LSTM モデルで学習を行う．LSTM 出力部では学習

表 1　追加する命令

種類	命令名	説明
行動の種類	else	直前の命令が実行できない場合以下の行動を行う
アクション	loop, STAGEX, Y	STAGEX からここまでを Y 回繰り返す
アクション	trick,PX	プレイヤー X のトリック数を 1 増やす
アクション	loss,PX,Y	プレイヤー X の持ち金をを Y 減らす

したモデルを使用して，STAGE0 の内容を入力（以下，初期シードと呼ぶ）として，トリックテイキングゲームの文章生成を行う．

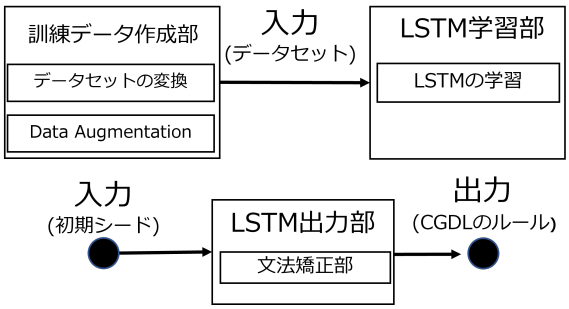

図 2　システム構成図

4.1　訓練データ作成部

4.1.1　データセットの変換

ここでは，CGDL で記述したトリックテイキングゲームのルールを，どのような形で学習させたかを述べる．本手法では以下の条件でルールを記述している．以下のルールでゲームの記述を行い，20 単語で 1 つのデータ（以下，バッチと呼ぶ）としてまとめる．

- 各ルールの文頭と文末に始まりと終わりを意味する‘|’を 20 個付ける
- stage の区切りには＜ ST ＞をつける
- データ中のカンマはスペースに置き換える．
- それぞれの単語には対応する番号があり，すべての単語を対応する数字に変換してデータとする．

4.1.2　Data Augmentation

本論文ではデータ数を増やすために Easy Data Augmentation(EDA)[9] を使用して，データの拡張を行った．本手法では，パラメータを以下の通りに設定している．

$$a = 0.05$$

$$n_{aug} = 16$$

編集確率を a，ワード移動を行う回数を n の値で設定している．パラメータは EDA の論文内で使われているものを採用した．

4.2 LSTM 学習部

LSTM 学習部では訓練データ作成部で受け取ったデータセットを入力として LSTM モデルを使用した深層学習を行う．計算設定を表 2 に示す．入力層を Input 層，出力層を Dence 層，中間層を Embedding 層と LSTM 層からなるニューラルネットワークを構築した．入力層では，［32 , 20］のサイズの系列データからなるテンソルを出力する．32 と 20 はそれぞれ，バッチサイズと seq_length の値である．Embedding 層では，［32，20，100］のサイズのテンソルを出力し，単語を変換した整数トークンを連続するベクトルに埋め込みそれぞれの単語の表現を学習する．100 の値は embedding_size の値である．LSTM 層では，［32，256］のサイズのテンソルを出力し，再帰的な学習を行う．Dence 層では，［32，63］のサイズのテンソルを出力し，各単語が出力される確率を求める．

表 2 計算設定

パラメータ名	ステータス
batch_size	32
seq_length	20
epoch	50
embeddingsize	100
活性化関数	softmax

4.3 LSTM 出力部

LSTM 出力部では学習をした LSTM モデルを使い，初期シードを入力としたルールの出力を行う．出力の構造について図 3 に示し，計算設定を表に示す．入力した文の次に来る単語を予測し，予測した単語を入力に加えて繰り返す．これをデータの始まりと終わりを示す‘|’が出るまで続ける．

4.3.1 文法矯正部

CGDL では特定の先行詞やアクションの後には，その対象となる値が必ず来るような厳格なルールがある．そこで生成の精度を上げるため，各単語の確率が出力された後に文法としてそぐわない単語を排除し，100%になるように確率の再計算を行う．例えば，指定した場所からカードを引くアクション

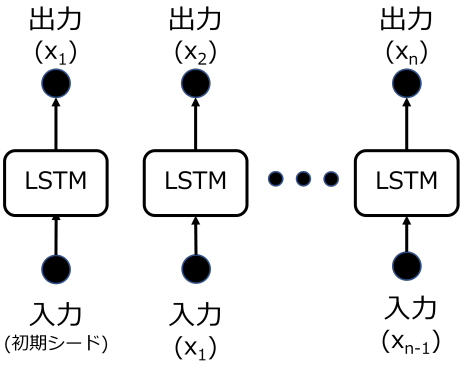

図 3 LSTM による文章生成

‘pifr’の後には必ずどこから引くのかを表す単語が入るので，場所を表す単語以外の要素を出力から排除する

4.4 生成されたゲーム

以下に生成した CGDL のゲームの一例を記す．

- STAGE0
  各プレイヤーにカード 7 枚と賭け金 99 を配る．
- STAGE1
  役の宣言を行い最も高い役を宣言したプレイヤーを決める．
- STAGE2
  各プレイヤーがカードを出す．その際に，最初のプレイヤーが出したカードと同じマークのカードがあるならば出さなければならず，ないなら好きなカードを出してよい．そして最も強いカードを出したプレイヤーの取得トリック数を追加する．
- STAGE3
  STAGE2 の内容を 5 回行う
- STAGE4
  STAGE1 で一番多くトリック数を宣言したプレイヤーの取得したトリック数が宣言した役より多い場合，宣言した役の得点を得て，少ない場合に得点を失う．

```
STAGE0
com_deal, allplayers, 7
com_give, allplayers, 99
STAGE1
mandatory_unconditional_bet, ram, ram
unconditional_bet, >, ka
once_unconditional_done
STAGE2
mandatory_show, ram, t1_playit
once_show, samesuit, tx_playit
else_show, ram, tx_playit
mandatory_play, t1, >, ta_trick p1
```

```
13 STAGE3
14 unconditional_loop, STAGE2, 5
15 STAGE4
16 mandatory_tokens, k12, >, p1_gain, ka1
17 else_loss, p1, k12
```

5 評価・考察

大学生 11 人を対象に，4.4 節で生成したルール（以下，ゲーム A とする）とトリックテイキングゲームの一つであるミニブリッジをプレイしてもらいアンケートに回答してもらった．ただし，CGDL は RANKING，STAGES，WINNINGCONDITIONS で構成されており，4.4 節で出力したのは STAGES のみであるため，ゲーム A に STAGES 以外を追加した．ミニブリッジのルールは日本コントラクトブリッジ連盟が公開しているルール [10] に従って行った．

5.1 ゲーム A

追加したルールは以下の通りである．追加したルールは生成されたゲームと類似しているカードゲーム"ナップ"から取っている．

- 使用するカードは 7~A までの 32 枚
- A が最も強く，7 が最も弱い
- 最もスコアが高いプレイヤーが勝利する．
- 役を宣言し，宣言した役を達成できた場合に点数を得る．失敗した場合点数を失う．役と得点の表を表 3 に示す．

表 3 得点表

説明	成功時の得点	失敗時の失点
2 回以上勝つ	2	2
3 回以上勝つ	3	3
全て負ける	3	3
4 回以上勝つ	4	4
全て勝つ	10	6

7 枚のカードを使って 5 回数字を比較するトリックを行い何回勝つかを予想するゲームである．じゃんけんで決めたプレイヤーから順番にすでに宣言されている役より得点が高い役を宣言し，最も高い役を宣言したプレイヤーは親になり 5 回トリックを行う．親の処理した回数が最初に宣言した役以上ならば親は点数を獲得し，少なければ失点するゲームである．最初の 1 回は親からカードを出し，残りのゲームはトリックを勝利したプレイヤーが最初にカードを出す．また，最初のプレイヤーが出したマークと同じマークがあれば必ず出さなければならず，ないならばどのカードを出してもよいがそのトリックは敗北する．

5.2 評価手法

11 人を 3 グループに分けミニブリッジ，ゲーム A を遊んでもらった．回答してもらったアンケートの結果を表 4 に．アンケートは 5 段階評価で 1 が最も悪く 5 が最も良い．3，4 つ目の質問はゲーム A にのみ行った．

表 4 評価結果（平均）

説明	ミニブリッジ	ゲーム A
面白かったか	4.55	4.55
駆け引きはあったか	4.36	3.91
4.4 節のルールとゲーム A は矛盾はあるか		3.64
ゲーム A は整合性がとれていたか		4.55

5.3 考察

ゲーム A は面白かったかという質問に対してはミニブリッジと遜色のない結果が出せており，ルールの整合性などの部分でも高い評価であることから，トリックテイキングゲームの手順は生成できていると考えられる．一方で，駆け引きはあったかという点に関しては差がでてしまった．理由として STAGES のみの学習では，駆け引きの部分まで完全に学習しきれていないのではないかと考えられる．

6 今後の課題

本手法では CGDL を用いたカードゲームの自動生成を試みた．結果，問題のないルールの生成という点では高い評価を得たがゲームの駆け引きでは芳しくない評価だった．以下に精度を上げるための今後の課題を示す．

- データ数の増加
  今のデータ数では様々なルールを表現するのが難しいため，トランプゲーム大全全てのゲームをデータセットにしたい．
- CGDL の見直し
  ルールの表現方法，駆け引きといった要素をどう表現するかを見直す．
- WINNING，RANKING の追加
  データセットにラベルのような形で，どのような要素があるのか表現したい．

参考文献

[1] Vincent Hom, Joe Marks . "Automatic design of balanced board games."
Proceedings of the Third AAAI Conference on Artificial Intelligence and Interactive Digital EntertainmentJune 2007 Pages 25–30

[2] 有田 隆也 (2011) 「ドイツボードゲームの教育利用の試み―考える喜びを知り生きる力に結びつける―」, コンピュータ＆エデュケーション 31(0), 34-39, 2011

[3] 松本太一 (2018) アナログゲーム療育ーコミュニケーション力を育てるー (幼児期から学齢期まで) ぶどう社

[4] 日本にボードゲームが広まらない理由
"https://tgiw.info/2011/08/post_1064.html"

[5] Jose M. Font , Tobias Mahlmann, Daniel Manrique1, Julian Togelius(2013) "A Card Game Description Language"

[6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory."
Neural computation 9.8 (1997): 1735-1780.
16th European conference on Applications of Evolutionary Computation

[7] 牧野 貴斗　濱川 礼 (2021) CGDL を用いたトリックテイキングゲーム自動生成手法の提案
研究報告ゲーム情報学（GI）2021-GI-46 巻 7 号

[8] 赤桐 裕二 (2014) トランプゲーム大全　スモール出版

[9] Jason Wei, Kai Zou(2019) "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks"
EMNLP-IJCNLP 2019 short paper

[10] 日本コントラクトブリッジ連盟 公式ホームページ "LET'S PLAY BRIDGE"
"https://www.jcbl.or.jp/Portals/0/pdf/fukyu/tools/letsplay1912.pdf"

7 付録

付録として STAGES の命令集を記載する．CGDL は，ゲームのルールを RANKING と STAGES, WINNINGCONDITIONS の三つで要素で記述した言語である．本論文では，CGDL の STAGES の生成を行っている．STAGES はゲーム中に行う行動を命令で表現している．命令は，行動の種類，先行詞，アクションの三種類の命令を組み合わせて構成されている．以下の表でそれぞれの命令の効果を記載する．

行動の種類	効果
once(1 度きり)	STAGE 中 1 回のみ行える命令
com(computer)	STAGE の開始時に行われる行動
mandatory(必須)	STAGE の開始時に必ず行う命令

先行詞	効果	使用例
tokens, KA, 制限, KB	トークンを置く場所 KA と KB のトークンの量を比較する． 制限を満たしている場合，アクションを行う	tokens, K01, ＞, K11
play, LA, 制限, LB	カードを置く場所 LA と LB のカードの強さ (役) を比較する． 制限を満たしている場合，アクションを行う	play, H0, ＜, H1
sum, LA, 制限, LB	カードを置く場所 LA と LB のカードの合計値を比較する． 制限を満たしている場合，アクションを行う	sum, H0,=,H1
have , 組み合わせ	プレイヤーが手札にカード，マーク，特定の数字の組み合わせ (連番など) を持っているか確認する． 持っている場合アクションを行う	have, 2 of diamonds
draw	デッキからカードを 1 枚引き， それを現在のプレイヤーの手札に加える． 加えたのちアクションを行う．	draw
show, 制限, LA	カードの場所 LA の一番上に置かれているカードと制限を 満たすカードを指定したのちアクションを行う．	show, card with samesuit, TO
λ (LAMBDA)	条件なしでアクションを行う	unconditional

アクション	効果	使用例
pifr, LA * 枚数, 向き	カードを置く場所 LA から枚数分指定 した向き (表，裏向き) にカードを引く	pifr, D*2, down
puin, LA * 枚数, 制限, 向き	カードを置く場所 LA に制限を満たす カードを指定した枚数分置く	puin, T0 +1, ＞, up
bet, 制限, KX	トークンを置く場所 KX のトークンの量と比較 したときに制限を満たすトークンの量を賭ける．	bet, ＞, K01
gain , KX	トークンを置く場所 KX のトークンを取得する．	gain, K21
playit	Antecedent で指定 (show) したカードを指定した場所に置く	-
next	プレイヤーの状態を next(行動済み) に変更する． next になったら次の順番が来るまで待機する．	-
done	プレイヤーの状態を done(行動済み) に変更する． done になったらそのステージ中行動できない．	-
out	プレイヤーの状態を out(脱落) に変更する． out になったらゲームから脱落する．	-
win	プレイヤーが勝利し，ゲームが終了する．	-
end	ゲームを終了する．	-