LLM と RoBERTa に基づくテキストゲームエージェントの実装と性能向上

鳥取大学
自然言語処理研究室
ZHUO BINGGANG

テキストゲーム

テキストゲームの特徴：
プレイヤーの行動とゲームのフィードバックが文字形式

あなたは汚れたティッシュを持っている。部屋にはゴミ箱がある。

> 汚れたティッシュをゴミ箱に入れる
汚れたティッシュをゴミ箱に入れた！1ポイント獲得！

ビジョンゲーム

テキストゲーム

研究背景と目的

テキストゲームエージェント

- テキストゲーム「を作る」研究ではなく、テキストゲーム「を解く」研究
- 自然言語処理と強化学習の接点に位置する重要な研究領域

研究目標

- 従来手法の課題を明らかにし、改善手法を提案
- より高いタスク達成率を安定的に達成できるエージェントの構築を目指す

研究の全体像：2種類のテキストゲーム環境

TWC (TextWorld Commonsense)

- 部屋を整理するゲーム
- 基本、常識能力を問う

FTWP (First TextWorld Problems)

- 料理するゲーム
- より複雑、指示を従う能力、安全性を重視

環境の多様性を反映

基本から複雑へ、段階的に研究を推進

研究の全体像：2種類の実装アプローチ

大規模言語モデルベース

- APIで呼び出す
- 訓練が不要
- 実装は簡単、最初の試行案として適切

ローカルモデルベース

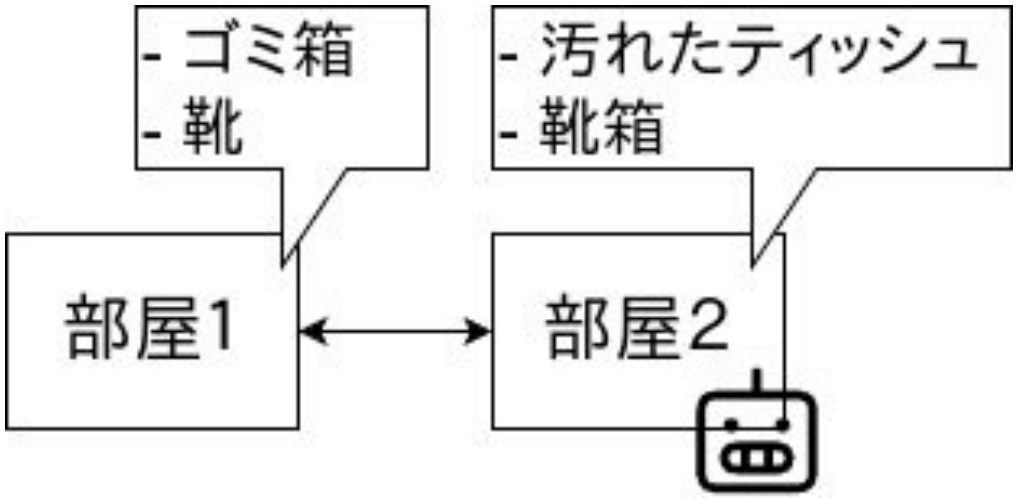
- ファインチューニング
- 実装はより複雑
- 性能がより安定、拡張性も高い

異なる環境に対し、エージェント実装の難易度、性能とAPIコストを総合的に考慮

TWCデータセットにおける研究

データセット紹介

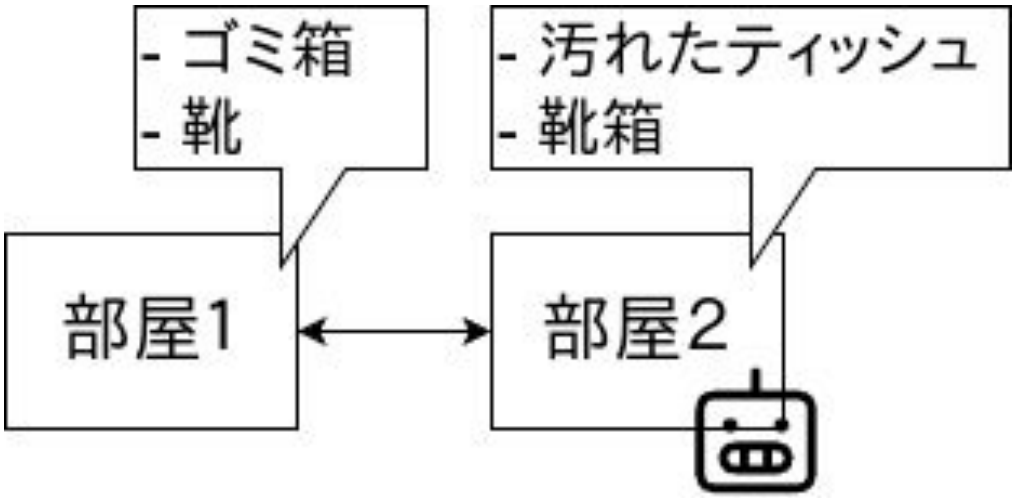
IBM researchが開発したTWCという環境を使用



エージェントの常識能力（何をどこへ置くべき）を問う

エージェントが所在部屋の状況しか把握できない

TWCの流れ



このゲームを解く行動例:

1. 汚れたティッシュを拾う
2. 西へ移動
3. 汚れたティッシュをゴミ箱に入れる
4. 靴を拾う
5. 東へ移動
6. 靴を靴箱に入れる

TWCにおける先行研究手法

既知最良手法：TWC Agent（強化学習＋知識グラフ）

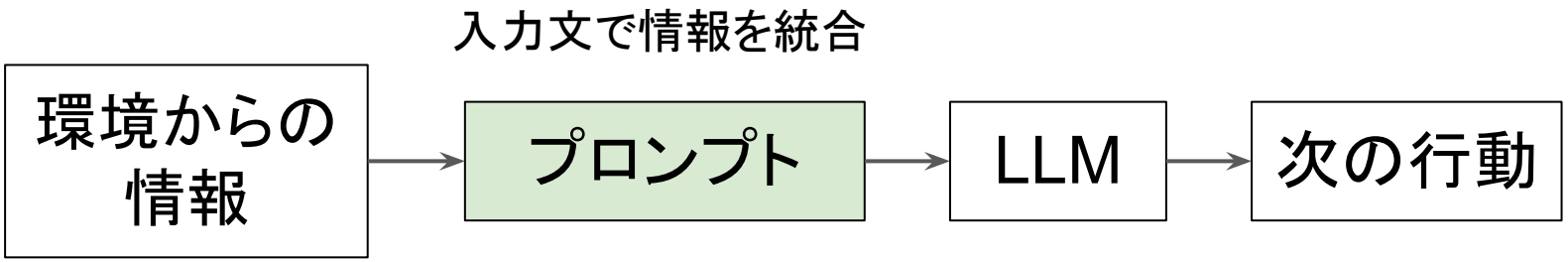
問題点

- 性能が低い (Hardレベルで57%のタスク完成率)
- 設計パラダイムが旧い (近年の技術未使用)

対応

- 近年急速に発展したLLMを活用

提案手法



プロンプト(入力文)設計が手法の基本

プロンプト設計のチャレンジ:

- エージェントにタスクを理解させる
- エージェントの記憶を維持
- エージェントの思考を促す
- 情報の質を確保する同時に入力長(コスト)を最小化
- …

提案手法：プロンプト設計

タスク：タスクについての説明
行動履歴：行動記録、エージェントの記憶
インベントリ：所持品リスト
現在の環境：見える環境の説明、ゲームから直接取得
可能な行動：リストをゲームから直接取得
考え：<fill in>
次の行動：<fill in>

プロンプト例

タスク: …あなたの目標は、物を適切な場所に配置してスコアを向上させることである。

行動履歴: action 0: ナイトスタンドから青いモカシンを取る -> モカシンを取った。action 1: 青いモカシンをワードローブに入れる -> モカシンをワードローブに入れた。

インベントリ: 何も持っていない。

現在の環境: -= 廊下 =- 開いた靴箱が見える。靴箱にはきれいな白い靴下が1組入っている。部屋を見渡すと帽子掛けがあるが、何もかかっていない…

可能な行動:

- 靴箱を閉める
- …

考え: <fill in>

次の行動: <fill in>

靴の一種

プロンプト例

タスク:…あなたの目標は、物を適切な場所に配置してスコアを向上させることである。

行動履歴:action 0:ナイトスタンドから青いモカシンを取る -> モカシンを取った。action 1:青いモカシンをワードローブに入れる -> モカシンをワードローブに入れた。

インベントリ:何も持っていない。

現在の環境:-= 廊下 =- 開いた靴箱が見える。靴箱にはきれいな白い靴下が1組入っている。部屋を見渡すと帽子掛けがあるが、何もかかっていない…

可能な行動:

- 靴箱を閉める
- …

考え:<fill in>

次の行動:<fill in>

ゲーム環境から取得した情報

手法：Output Template

タスク：…あなたの目標は、物を適切な場所に配置してスコアを向上させることである。

行動履歴：action 0：ナイトスタンドから青いモカシンを取る -> モカシンを取った。action 1：青いモカシンをワードローブに入れる -> モカシンをワードローブに入れた。

インベントリ：何も持っていない。

現在の環境：-= 廊下 =- 開いた靴箱が見える。靴箱にはきれいな白い靴下が1組入っている。部屋を見渡すと帽子掛けがあるが、何もかかっていない…

可能な行動：

- 靴箱を閉める
- …

考え：<fill in>

次の行動：<fill in>

出力統一のためのプロンプト設計技術：Output Template

手法：Chain-of-Thought

タスク：…あなたの目標は、物を適切な場所に配置してスコアを向上させることである。

行動履歴：action 0：ナイトスタンドから青いモカシンを取る -> モカシンを取った。action 1：青いモカシンをワードローブに入れる -> モカシンをワードローブに入れた。

インベントリ：何も持っていない。

現在の環境：-= 廊下 =- 開いた靴箱が見える。靴箱にはきれいな白い靴下が1組入っている。部屋を見渡すと帽子掛けがあるが、何もかかっていない…

可能な行動：

- 靴箱を閉める
- …

考え：<fill in>

次の行動：<fill in>

深い思考を促すプロンプト技術：Chain-of-Thought

手法：One-Shot Prompting

タスク：…あなたの目標は、物を適切な場所に配置…

サンプル：action0：汚れた黄色のドレスを洗濯機に入れる。-> 汚れた黄色のドレスを洗濯機に入れた。スコアが1ポイント上がった…… action14：汚れたマルーンのドレスを洗濯機に入れる。-> 汚れたマルーンのドレスを洗濯機に入れた。スコアが1ポイント上がった。

行動履歴：action 0：ナイトスタンドから青いのモカシン…

インベントリ：何も持っていない。

現在の環境：-= 廊下 =- 開いた靴箱が見える…

可能な行動：

- 靴箱を閉める
- …

考え：<fill in>

次の行動：<fill in>

合理的な行動を促すために、一つの行動例を提示する技術：One-Shot

手法：フィードバック増強（提案手法）

フィードバック増強(Feedback Augmentation, FA)とは　シンプルなオリジナルプロンプト技術

行動履歴：action 0：ナイトスタンドから青いモカシンを取る -> モカシンを取った。action 1：青いモカシンをワードローブに入れる -> モカシンをワードローブに入れた。間違い場所である。action2: 青いモカシンを取る。-> モカシンを取った。action3: 青いモカシンを靴箱に入れる -> モカシンを靴箱に入れた。スコアが1ポイント上がった。正しい場所である。

追加記述でフィードバックを明確にすることでより合理的な行動を促す

実験

評価基準：normalized score

- 達成したスコア ÷ 達成できる最大スコア
- 結果範囲は0~1、タスクの達成率を表し

データセット情報：3つの難易度

難易度	部屋数	整理必要な物品数
Easy	1	1
Medium	1	2~3
Hard	1~2	6~7

実験結果

	Easy	Medium	Hard
LSTM-A2C	0.86	0.74	0.54
DRRN	0.81	0.73	0.44
KG-A2C	0.85	0.72	0.46
TWC-Agent	0.96	0.85	0.57
提案手法	1.00	1.00	0.70

- 提案手法の性能が先行手法を上回る
- EasyとMediumレベルでのタスク完成率は100%
- Hardレベルでは70%のタスクを完成

アブレーション実験結果

	Easy	Medium	Hard
提案手法	1.00	1.00	0.70
- GPT-4	1.00	0.58	0.18
- CoT(Chain of Thought)	1.00	1.00	0.61
- One-Shot	1.00	1.00	0.64
- FA (フィードバック増強)	1.00	1.00	0.52

- LLMの性能が肝心：GPT-4をGPT-3.5-turboに置き換えると、性能が大幅に低下（Hard: 0.70 → 0.18）
- プロンプト設計が性能に寄与：いずれかのプロンプト技術を除去すると、性能が低下
- フィードバック増強が有効：フィードバック増強を行わない場合、性能が大きく低下（Hard: 0.70 → 0.52）

より複雑なゲーム環境へ

LLMに基づくエージェントの問題点

- TWCよりも複雑なゲーム環境を対応できない可能性
- **APIの使用コスト**

今後の方向性

- より複雑なゲーム環境に対応するために、ローカルモデルを微調整するアプローチが必要

より複雑なゲームへ: ローカルモデルの性能比較

ローカルモデルの選定

本研究の目的：テキストゲームエージェントの基盤として適切なローカルモデルの選定

評価基準：自動強調付与タスクにおける性能

理由：

- テキストゲームを解くには、言語理解力が重要
- 文章の重要箇所が分かる → 言語理解力が高い

それゆえ、自動強調付与タスクをベンチマークとし、ローカルモデルの言語理解力を検証

自動強調付与データセット

ライフハッカー

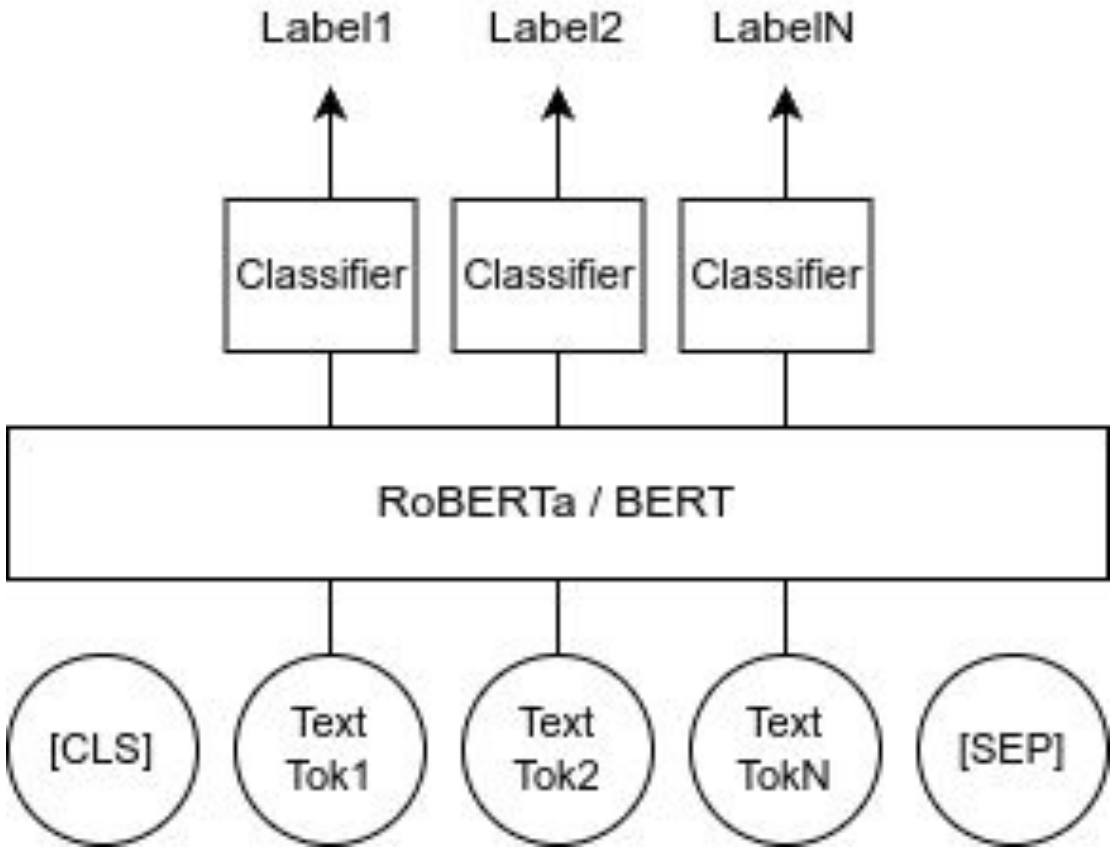
- Lifehacker : 日本のライフスタイル系ウェブサイト
- ウェブサイトから331本の記事を収集
- 太字を正解ラベルに
- 5分割クロスバリデーションで性能検証

ローカルモデルの候補

fastText + BiLSTM	fastTextとは事前学習された単語モデルの一種、word2vecやGloVeよりも優れた性能を持つ
BERT	事前学習されたTransformerモデルの一種、言語理解タスクに得意
RoBERTa	BERTの改良版であり、BERTより良い性能が報告される

Transformer：近年登場する深層学習モデル構造の一種、言語理解タスクに強い性能を示す

Transformerモデルで自動強調付与



- 強調レベルは単語
- 入力文を単語に分け、モデルにタグを付与させる

性能評価結果

	適合率	再現率	F値
CRF(先行研究)	0.314	0.312	0.313
fastText + BiLSTM	0.277	0.324	0.297
BERT	0.376	0.372	0.362
RoBERTa	0.372	0.444	0.399

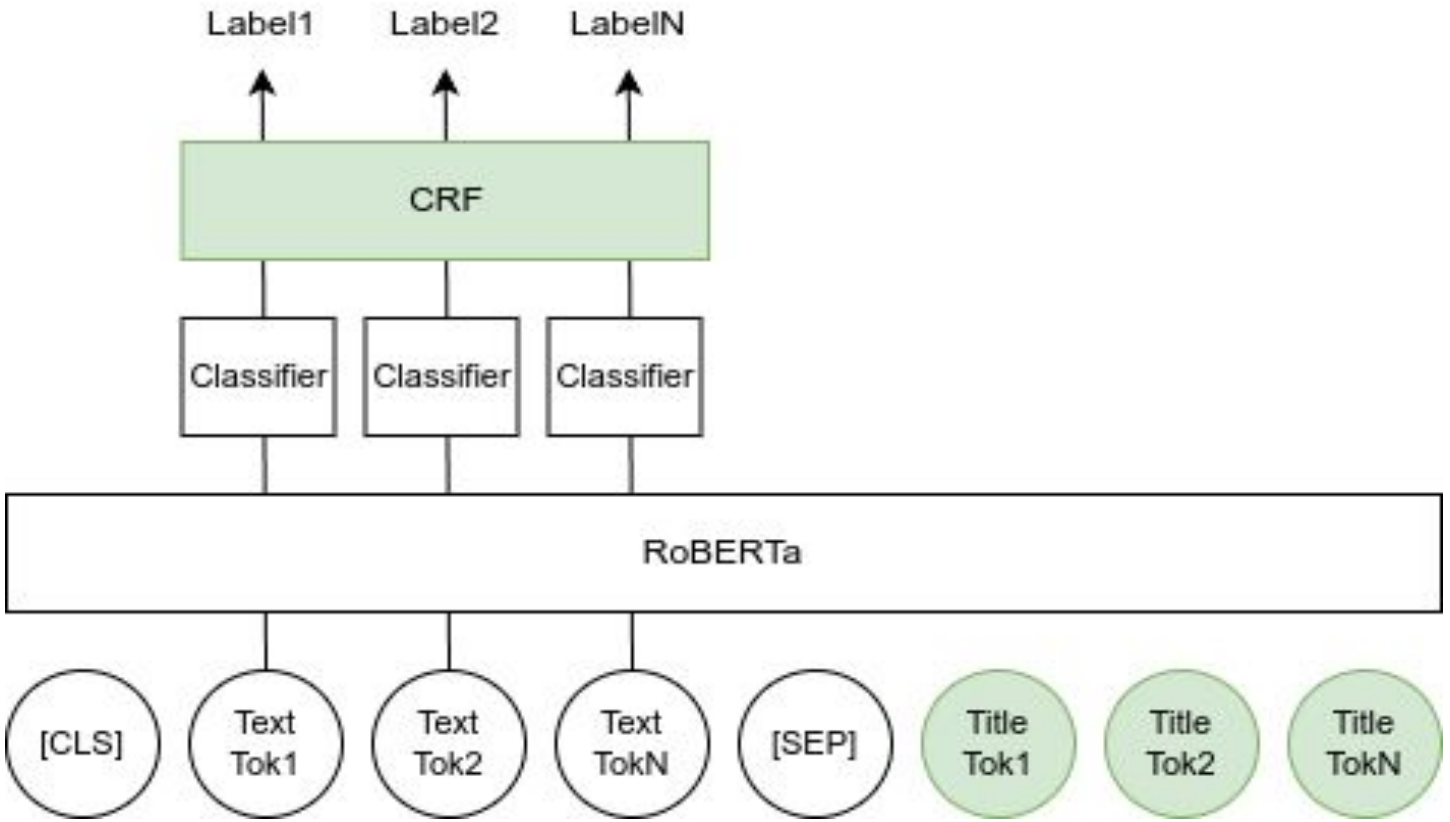
RoBERTaはF値で他の手法を大きく上回り、強い言語理解力を示唆

この結果に基づき、**以降のテキストゲーム研究では一貫してRoBERTaを基盤モデルとして使用**

さらなる性能向上

RoBERTaを基盤として、手法の改良を試みた:

- タイトル情報を活用して強調を補助
- 出力の一貫性を高めるためにCRF層を導入



実験結果

	適合率	再現率	F値
BERT	0.376	0.372	0.362
RoBERTa	0.372	0.444	0.399
改良RoBERTa	0.398	0.509	0.437
- CRF	0.414	0.453	0.423
- Title	0.360	0.476	0.401

- F値は0.399から0.437へと大きく向上
- CRF層やタイトル情報を使わないと性能が下がる
- 特にタイトル情報を外すと影響が大きい

FTWPデータセットにおける研究

データセット紹介

Microsoft researchが開発したFTWPという環境を使用

テキストゲーム分野において**既知最大級** データセット

タスク：調理本の指示に従って調理

特徴：エージェントの**指示を従う能力** を重要視

- 調理本の指示と矛盾する行動でゲームオーバー

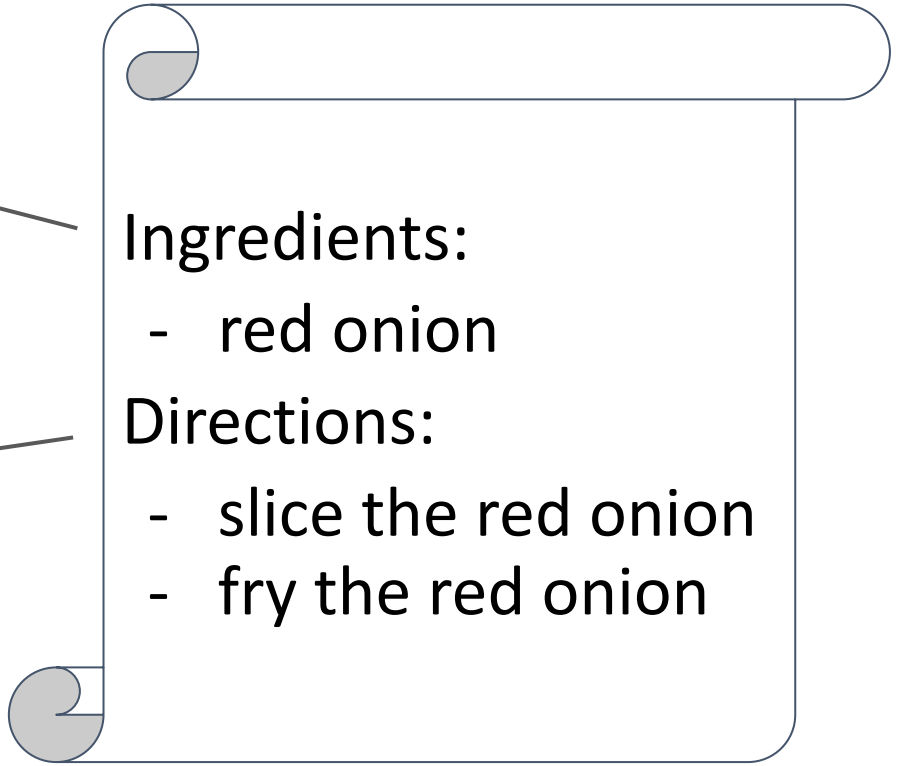
調理本の例

必要な食材:

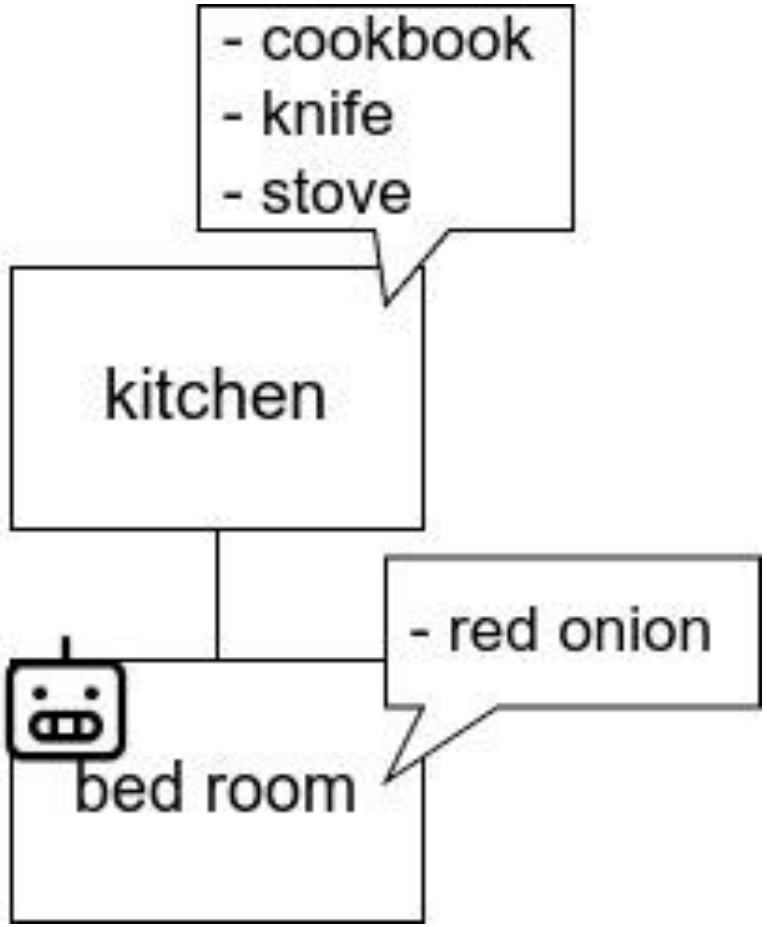
赤い玉ねぎ

調理法:

1. 赤い玉ねぎをスライス
2. 赤い玉ねぎを炒める



FTWPの流れ



行動例：

1. go north
2. examine cookbook
3. go south
4. take red onion
5. go north
6. take knife
7. slice red onion with knife
8. cook red onion with stove
9. prepare meal
10. eat meal

Ingredients:
- red onion

Directions:
- slice the red onion
- fry the red onion

誤り行動分類器で FTWP エージェントの性能·安全性向上

誤り行動

調理本の指示と矛盾する行動を誤り行動と呼ぶ

種類1：ナイフ関連の誤り行動

- ナイフの切り方が様々： slice、chop、dice
- 「slice the red onion」という指示に対し
  - ✓ slice red onion with knife
  - ✗ chop red onion with knife
  - ✗ dice red onion with knife

誤り行動が実行されるとゲームオーバー

- 一旦「chop」された食材はもう「slice」できず　（不可逆）、ゲームは失敗となる

誤り行動

種類2：調理器具関連の誤り行動

- ゲーム内には stove、oven、BBQ など、さまざまな調理器具が存在
- 指示に従って、適切な調理器具を選ぶことが必要

例えば、「fry the red onion」という指示に対して

- 正しい行動は「cook red onion with stove」
- ovenやBBQで調理すると、ゲームは失敗

誤り行動は性能の妨げ

FTWPデータセットにおける既知最良エージェントを再実装し、検証データセットでの失敗事例を分析

誤り行動の実行がゲーム失敗の主要因

- もう一つの原因は行動回数上限の超過

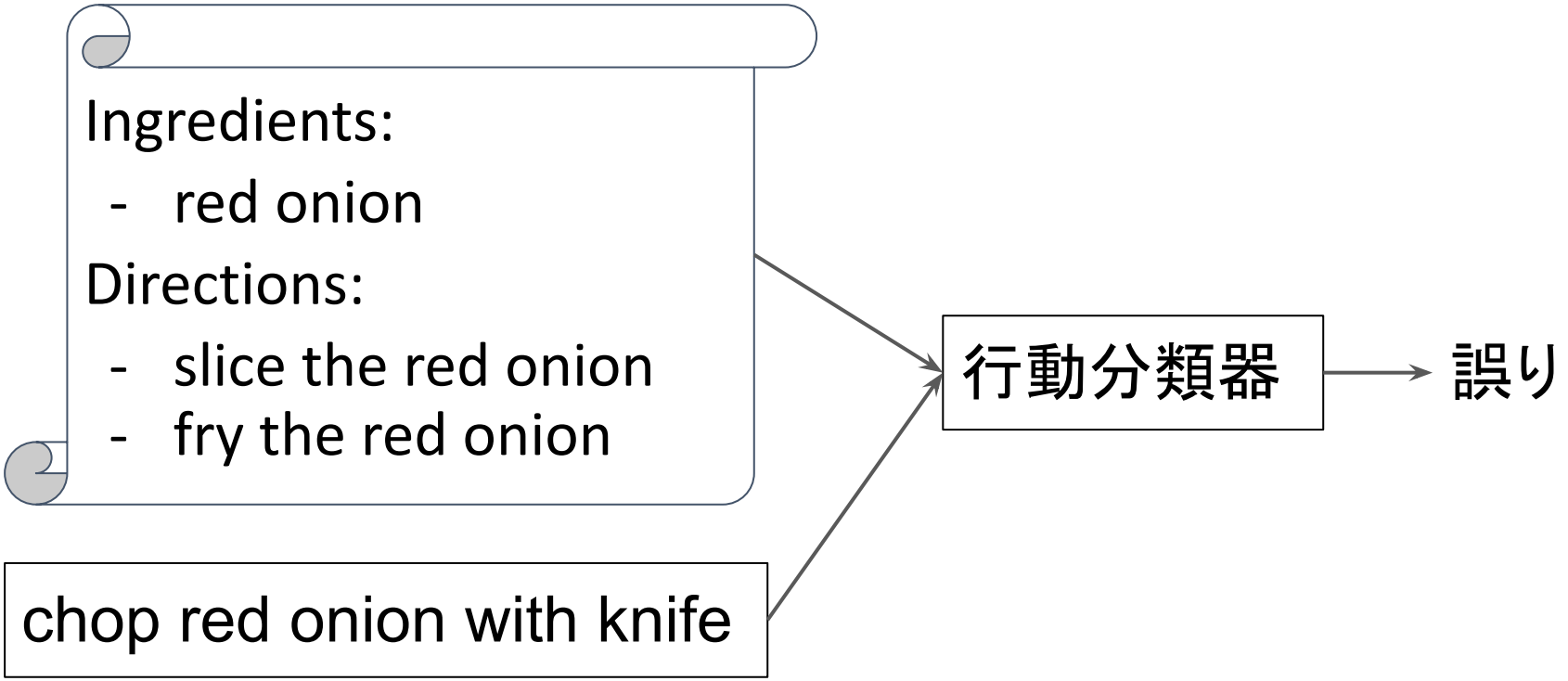
そこで、

誤り行動を減らす→性能向上

提案手法の構想

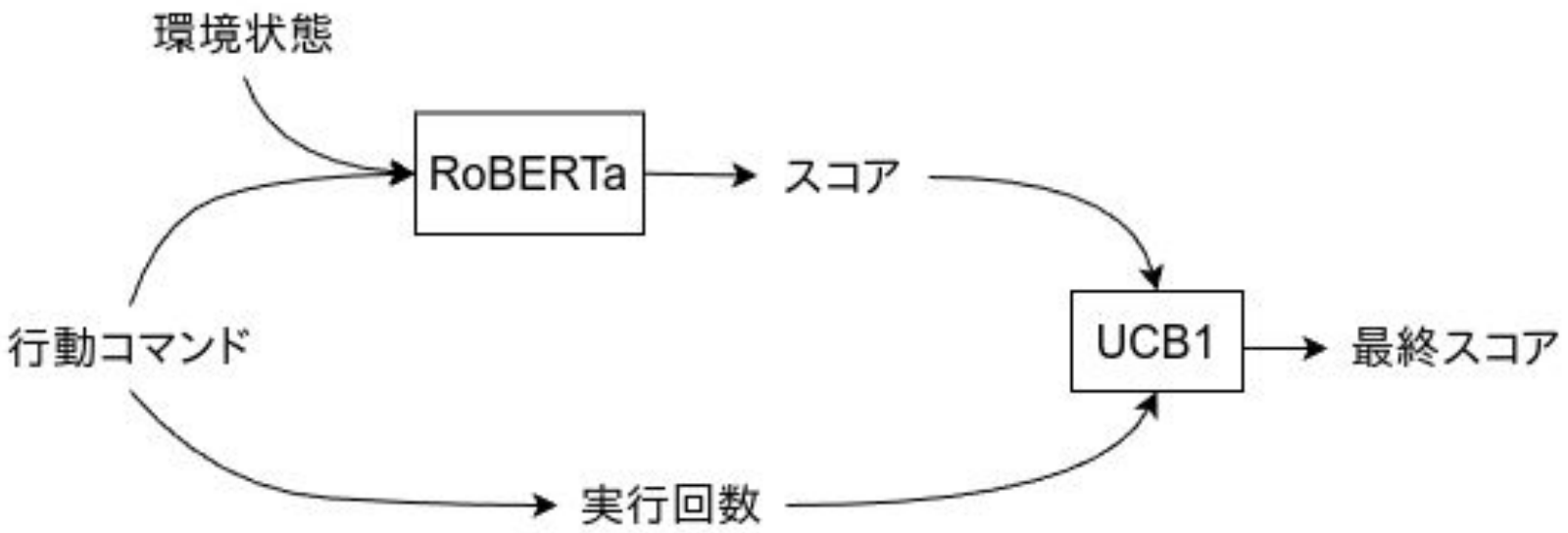
誤り行動の実行を減らすために

- 独立した誤り行動分類器を訓練
- 誤りと認識された行動の選択確率を低減



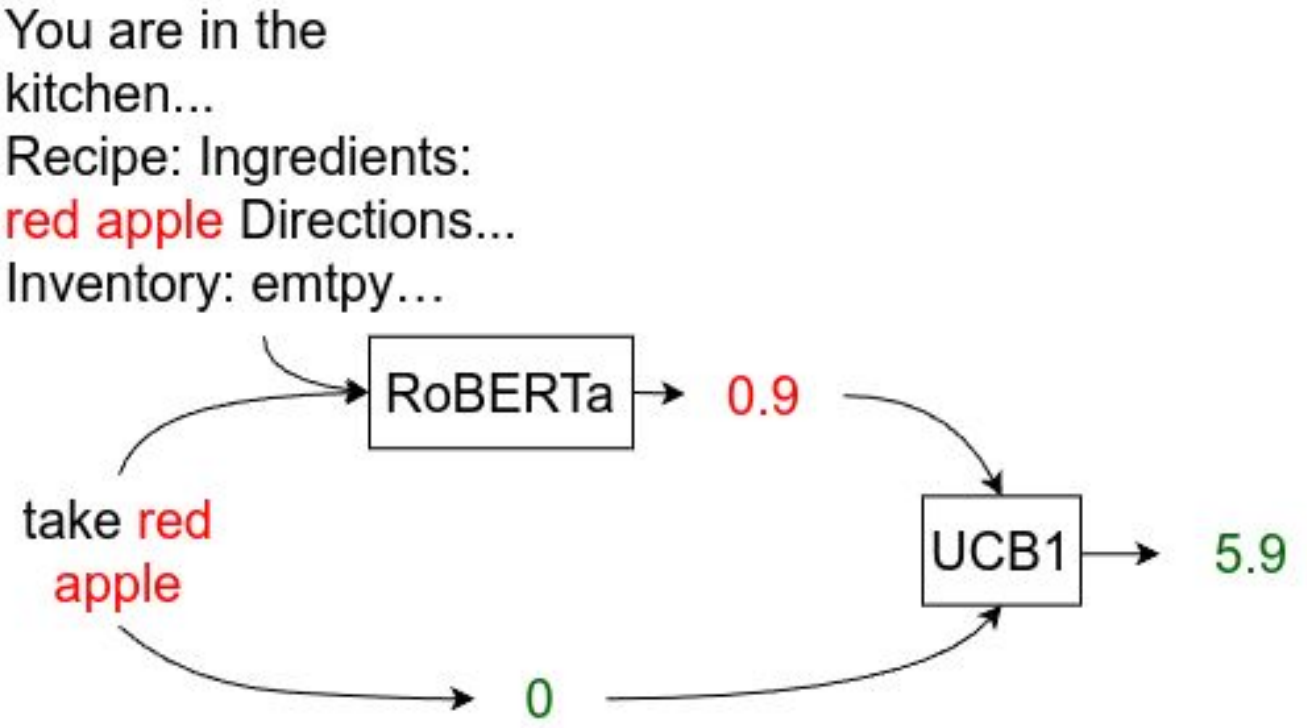
先行研究

CogniTextworldAgent：既知最良エージェント



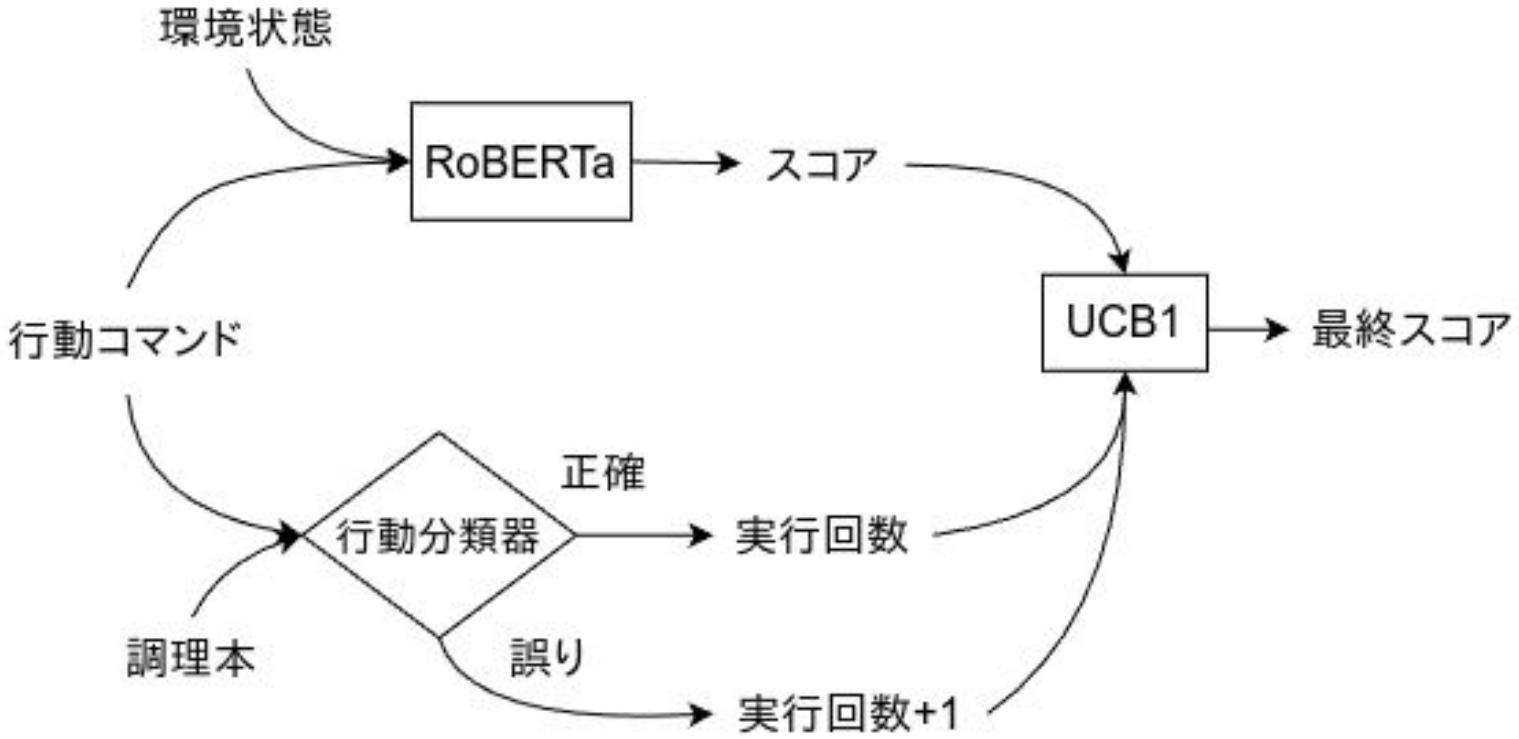
1. RoBERTaが行動にスコアをつける
2. UCB1アルゴリズムが未実行な行動を優先にし、無駄な繰り返しを回避

先行研究の入力例



- りんごが必要であるため、take red appleのスコアが高い(0.9)
- 実行回数が0であるため、UCB1が高いボーナスをつける(+5.0)

提案手法



- 誤り行動を完全に禁止するではなく、UCB1でそのスコアを抑制

実験結果

	Norm. Score	Avg. Steps
LeDeepChef	0.693	43.9
CogniTextworld-Agent	0.880	21.6
提案手法	0.912 (0.032↑)	24.7

- Norm. Scoreは、エージェントのタスク完成率
- LeDeepChefとは性能が二番目に高い先行手法
- 性能差が有意差あり（$p<0.05$）
- Avg. Stepsはゲーム終了までの平均行動回数、スコアが一定である場合ステップ数が少ないほうが効率が良い
- 平均ステップ数の増加原因：誤り行動による即ゲームオーバーを回避し、ゲームをより長くプレイ

実験結果：誤り行動の実行回数低減

	テストゲーム222個における誤り行動の実行回数
Agent 1	80 → 51
Agent 2	100 → 67
Agent 3	108 → 60
平均	96.0 → 59.3 (↓38.2%)

大きく低減した誤り行動の実行回数が誤り行動分類器の有効性と安全性を表す

ナビゲータによる FTWP エージェントの性能向上

FTWPゲームにおける探索の重要性

FTWPタスク

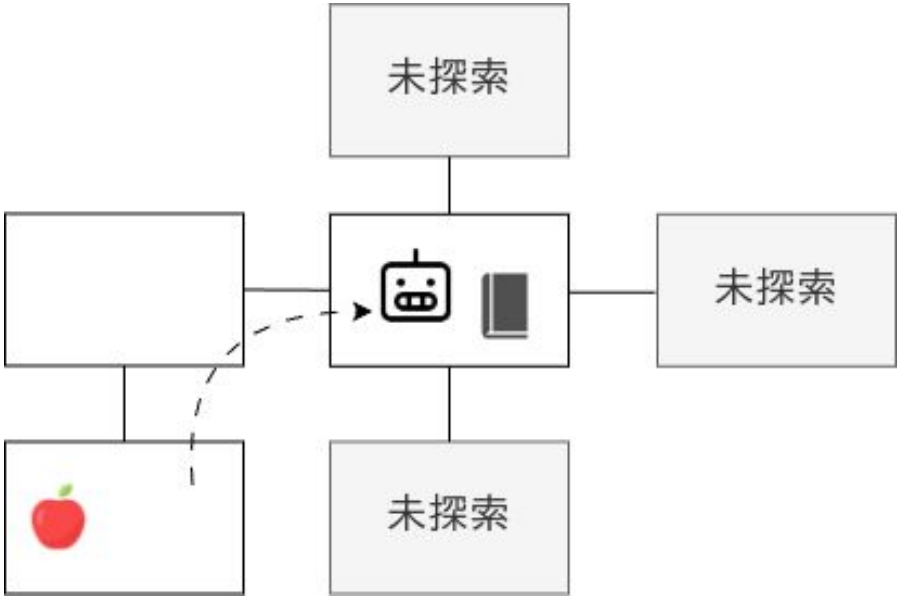
- 調理本(cookbook)を探す
- 調理本の指示に従い、食材を集める
- 調理器具を探し、食材を適切に処理する
- 作った料理を食べる

ポイント

- 探索はゲームの中心的要素：各部屋を移動しながら、調理本・食材・調理器具を探索
- 空間・物品の記憶力とナビゲーション能力が効率的なプレイに不可欠 → ナビゲータの提案

ナビゲータの構想

たとえば、
エージェントがキッチンで調理本をチェックし、
赤いリンゴが必要な食材だと分かる



一回見たことがあるので、
自動的に一気にリンゴのある部屋に移動することが望ましい → ナビゲータでこの機能を実現

ナビゲータの実装

目的：見たことのある物品の部屋に自動的に移動

仕組み

- エージェントの行動に伴いマップを記録・更新
- 食材と調理器具の位置を記録・更新
- エージェントが調理本を確認後、必要な食材のナビゲーションコマンドを提供

ナビゲーションコマンド

- 目的：一つのコマンドで一連の移動を実行
- 形式：navigate to <object>

ナビゲーションコマンドの提供

エージェントの行動リストに追加

- 行動リスト：ゲーム環境から貰える現時点で実行可能な行動のリスト

追加後の行動リスト例

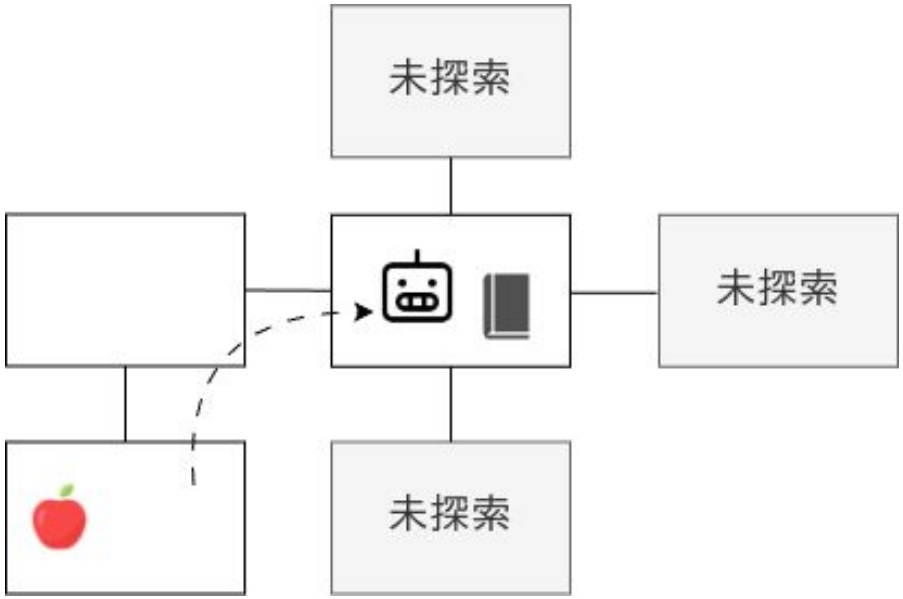
1. go east
2. go south
3. go west
4. go north
5. examine cookbook
6. navigate to apple

ナビゲーションコマンドの実行方法は次ページで紹介

ナビゲーションコマンドの実行

実行手順

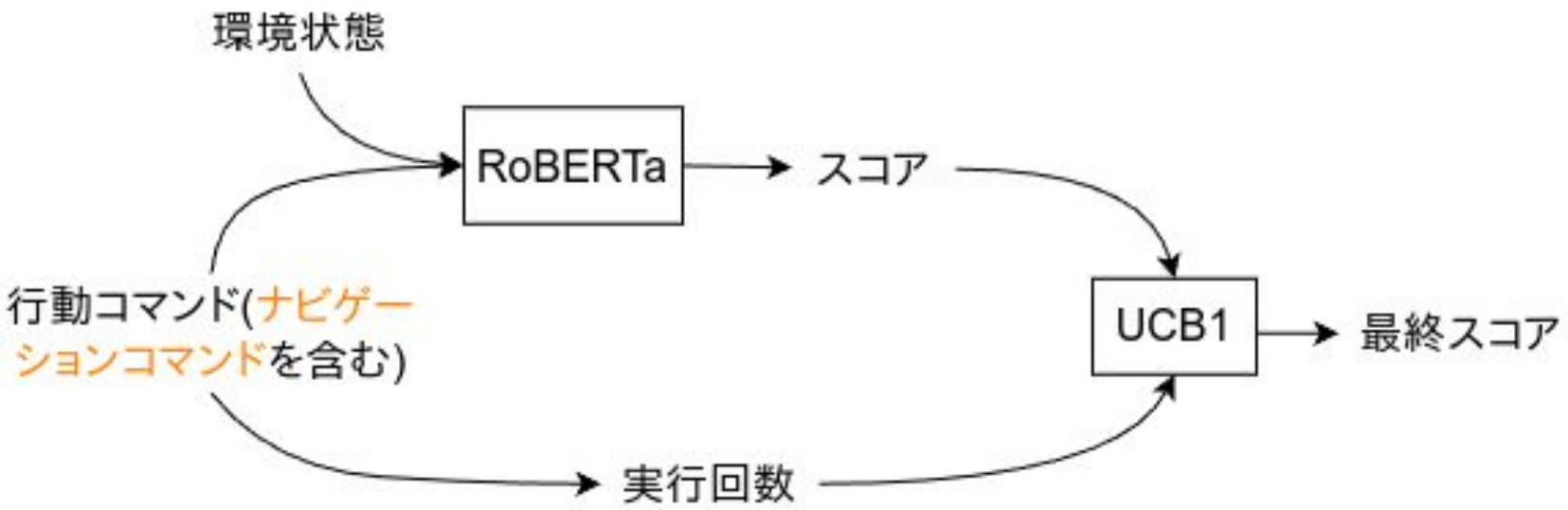
1. マップと物品記録をもとに移動経路を生成（BFS）
2. 移動経路をゲームシステムが認識できる移動コマンドに変換
3. 移動コマンドリストを順に実行し、目標物へ到達



navigate to appleの対応移動コマンドリスト:

1. go west
2. go south

エージェントの変化

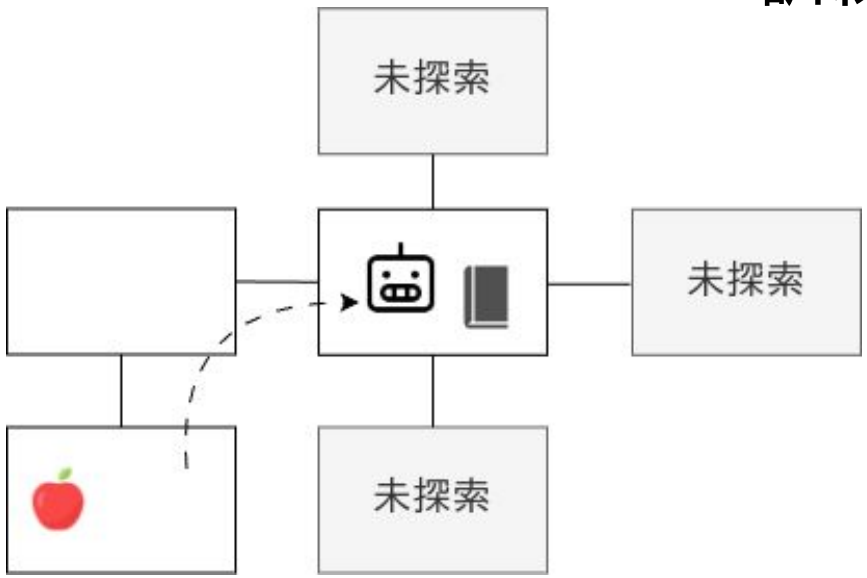


他のコマンドと同様、ナビゲーションコマンドにスコアがつけられる

しかし、訓練集に存在しないため、ナビゲーションコマンドのスコアが低い → 選択確率が低い

エージェントにナビゲーションコマンドを教える

- 訓練集にある移動コマンドをできるだけナビゲーションコマンドに変換



訓練集にあるゲームを解く行動例：

["inventory", "go north", "go east", "examine cookbook", "go west", "go south", "take apple", ...]

↓

["inventory", "go north", "go east", "examine cookbook", "navigate to red onion", "take apple", ...]

- 再訓練したエージェントが自発的にナビゲーションコマンドを使用

実験結果

	Norm. Score	Avg. Steps
LeDeepChef	0.693	43.9
CogniTextworld-Agent	0.880	21.6
＋　ナビゲータ	0.937 (0.057↑)	21.2
CogniTextworld-Agent + 行動分類器	0.912	24.7
＋　ナビゲータ	0.968 (0.056↑)	22.8

- ナビゲータで大幅に性能向上
- 性能差が有意差あり（$p<0.05$）

まとめ

データセット	環境特徴	実装アプローチ	提案手法の主張点
TWC	小規模・課題が単純・常識の活用が大事	LLM ＋ プロンプト設計	フィードバック増強
FTWP	大規模・多段階タスク・行動の安全性を重要視	ローカルモデル ＋ 拡張モジュール	● 誤り行動分類器 ● ナビゲータ

RoBERTaは強い言語理解力を示し、ローカルエージェントを実現するための有力な選択肢

今後の発展

模倣学習 → 強化学習へ

- 現段階：RoBERTaが専門家データを模倣
- 次段階：専門家データなしでも自律的に探索・学習できるエージェントへ

テキストゲーム → ビジョンゲームへ

- テキストゲームにおける研究成果をビジョンゲームに適用

付録

スライドと論文の対応関係

ページ	博士論文	外部論文
6-20	第3章	Binggang Zhuo, and Masaki Murata. “Utilizing GPT-4 to Solve TextWorld Commonsense Games Efficiently.” Proceedings of the 10th Workshop on Games and Natural Language Processing@ LREC-COLING, vol.2024. pp.76-84
22-29	第4章	ZHUO, Binggang, Ryota HONDA, and Masaki MURATA. “Information for Transformer-based Japanese Document Emphasis.” IEICE Transactions on Information and Systems (2025), vol.E108-D, no.7, pp.808-819
30-43	第5章	Binggang Zhuo, and Masaki Murata. “Enhancing Text Game Agent Performance via Wrong Action Classification.” Proceedings of the 2025 9th International Conference on Natural Language Processing and Information Retrieval, 掲載決定.
44-52	第6章	ZHUO, Binggang, and Masaki MURATA. “Enhancing Text Game Agent Performance via Navigator.” IEICE Transactions on Information and Systems, 投稿中.