GPTを用いたデスクトップ・エージェント・システムの開発

坂本 晋太郎　安藤 崇央

本研究では、ユーザとエージェント間の自然言語による対話を通じてユーザの望む GUI の自動操作を実現するデスクトップ・エージェント・システムを開発した。本システムでは、ユーザは音声認識を使用してエージェントと対話でき、エージェントは GPT-3.5 を用いてそのユーザの入力を解釈しタスクを実行する。エージェントによるタスクの実行は、ユーザが普段行う GUI 操作と同様の操作を自動的に行うことでなされる。これは、デスクトップ画面の画像解析等を利用し、デスクトップ上の適切な箇所へのマウスカーソルの移動や、キーボードからのキー入力を模倣することで実現している。

1　はじめに

近年、AI の急速な進化は大きな話題となっている。特に、GPT などの生成系 AI の発展により AI が人間に近い振る舞いをすることが可能になってきている。これにより、AI による人間とコンピュータのインタラクションについての研究は、様々な分野での応用が期待される。特に、AI を活用したエンターテイメントは大きな注目を集めると考える。

2023 年にリリースされた OpenAI [1] 開発の大規模言語モデル GPT-4 [2] は、非常に人間らしい文章の生成を可能とした。それだけでなく、米司法試験で上位 10 パーセントの成績を出すなど、専門的、学術的な能力も非常に高い。

また、スマートフォンやホームオートメーションシステムが普及し、音声 UI を用いた AI アシスタントは一般的になっている。しかし、現時点での AI アシスタントは、決められたタスクや対応しているソフトウェアに限られている。ユーザが行う GUI 操作を模倣してシステムがタスクを実行可能であればより多様な操作を実現できると考える。

そこで本研究では、音声認識、画像認識を大規模言語モデルと接続し、対話可能なキャラクターエージェントの開発を行う。特に、画像認識と GPT-3.5 を使用した汎用的なタスクの実行を可能とするシステムの実装を目的とする。加えて、エージェントに固有の記憶を持たせる記憶システムを実装し、ユーザとエージェントが円滑に対話できることを目的とする。

2　関連研究・技術

2.1　AI アシスタント

AI アシスタントまたは仮想アシスタントは、個人のタスクまたはサービスを実行するソフトウェアエージェントであり、iOS の Siri や Windows の Cortana、スマートスピーカーに搭載されている Amazon Alexa などがある。

これらは総じて、音声の認識、音楽再生、リマインダーの設定、インターネット上のウェブサイトからの情報を用いた質問の回答、各デバイスのアプリケーションの利用や制御が可能である。

2.2　GPT

GPT（Generative Pretrained Transformer）は、OpenAI が開発する高性能な言語モデルである。

Development of a Desktop Agent System Using GPT
Shintaro Sakamoto, Takahiro Ando, 群馬大学 理工学府 理工学専攻 , Department of Science and Technology, Graduate School of Science and Technology, Gunma University.

Transformer と呼ばれる深層学習の手法を用いた言語モデルで GPT-3.5、GPT-4（2023 年）などがある。文章の生成、文章の要約、質問への回答、翻訳などのタスクが可能である。

言語モデルとは、人間の言語を単語の出現確率でモデル化したものである。一般的な言語モデルは、大量のテキストデータを使って事前に学習したベースモデルを用意し、さらに、教師データを使用してファインチューニングを行うことで作成する。教師データとは、言語モデルの用途に応じた例題と答えのペアである。目的に対して高い精度を実現するような教師データの準備には多大な労力がかかる。最新モデルである GPT-3.5 や GPT-4 の具体的な学習内容は公開されていないが、GPT-3 の時点で、1750 億個のパラメータを持つといわれる [3]。このような桁違いに膨大なデータを用いて学習した後、人間のフィードバックからの強化学習を行うことで、ファインチューニングを必要としない言語モデルを確立した。本研究では GPT-3.5 の API を利用した。

2.3 Tesseract

Tesseract（テッセラクト）は、さまざまなオペレーティングシステム上で動作する光学式文字認識（OCR）エンジンである [4]。 Apache License の下でリリースされたフリーソフトウェアである。コマンドライン、または API を使用して画像から印刷されたテキストを抽出できる。英語や日本語を含む 100 を超える言語をサポートし、他の言語でも動作するようにトレーニングできる。

3 システム概要

本研究で開発したシステムの一連の動作は以下の通りで、これを模式図に表したものが図 1 である。

1. 本システムの UI から自然言語でタスク（GUI の操作）を命令する。
2. タスクを解釈し、GUI の操作を表す DSL に変換する。
3. DSL で記述された操作を実行する。
4. ユーザの入力と実行結果に対するエージェントの応答を UI に出力する。

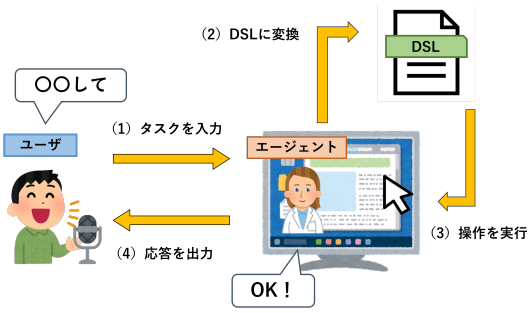

図 1　システムのイメージ図

図 2　システムの GUI

上記の仕様を満たすために本システムには以下の機能を実装した。

1. UI
2. GUI の操作を表す DSL
3. 自然言語のタスクを DSL のコードに変換する機能
4. DSL で記述された操作を実行する機能
5. エージェントの応答文を生成する機能

4 システム詳細

4.1 UI

Google の音声認識 API [5] とテキスト読み上げソフトウェアの VoiceVox [6] を使用して音声 UI を実装した。テキストを使用した入出力のために図 2 の GUI を実装した。

テキスト 1　DSL で記述した「Slack で送信」のコード

```
search,Slack
click,$送信する宛先
input,$送信する内容
press,ctrl-enter
```

4.2　DSL

本システムでは、タスクを複数の基本操作のシーケンスとして構成し、DSL を用いて定義する。実装した 6 つの基本操作と DSL の関係を表 1 に示す。

ユーザは任意のタスクを DSL で定義できる。例をテキスト 1 に示す。各操作の必要な情報は、$ 記号を接頭辞として記述することで入力に依存した値を設定可能である。

4.3　自然言語から DSL への変換

GPT-3.5 の Function calling API [7] を使用して自然言語の入力から適切なタスクを呼び出す。Function calling は GPT-3.5 の prompt に関数の定義を与えることで、自然言語の入力に対して、関数を実行するか否を判断し、実行する場合はその引数を返す機能である。本システムでは、定義した DSL のコードを$で記された箇所を引数として持つような関数とみなし、Function calling の prompt を設定した。図 3 の入力に対して、テキスト 2 の DSL のコードが得られる。

4.4　自動操作機能

DSL で記述された操作シーケンスを順に実行する。クリック操作の機能は画像解析を使用して実装した。デスクトップ画面のスクリーンショットに対して、語句の場合は Tesseract を用いて一致する語句の座標を検出する。画像の場合は openCV のテンプレートマッチングを用いて座標を検出する。Tesseract を用いた ORC の結果を図 4 に示す。

4.5　エージェントの応答文の生成

GPT-3.5 と記憶機能を利用してエージェントの応答文を生成する。記憶機能は、直前の対話履歴と過去

Slack で Shintaro にこれはテストですって送信して。

図 3　ユーザの入力

テキスト 2　Function calling の結果から作成した DSL のコード

```
search,Slack
click,Shintaro
input,これはテストです
press,ctrl-enter
```

図 4　Tesseract を用いた ORC の結果（赤枠が認識されたテキスト）

の対話データの要約を記憶する。また、入力を埋め込みベクトル [8] に変換し、記憶した過去の対話データの中でコサイン類似度が高いものを検索することで、入力に関連する過去の対話データを取得できる。GPT-3.5 の prompt に以下を設定する。

- エージェントの台詞例などの設定
- 入力に関連する過去の対話データ
- 直前の対話履歴
- 入力

4.6　常駐機能

以下の機能を実装し、ユーザのアクションがない場合もエージェントからユーザに向けて話題提供を行っている。図 5 のようなデータを取得した際、図 6 のように応答する。

表 1　基本操作と DSL の関係

基本操作	DSL の記法	必要な情報
クリック	click	クリックする語句/画像名
ダブルクリック	click2	クリックする語句/画像名
文字列入力	input	入力する文字列
キー入力	press	入力するキー
ウィンドウのアクティブ化	active	ウィンドウ名
スタートメニュー検索	search	検索する文字列

時刻：18:00 (水曜日), 気温：7°　C, 天気：雨

図 5　取得した天気予報

今は 18 時なのだ。気温は 7 ℃で少し寒いのだ。雨が降っていて、外出するのはちょっと難しいなのだ……。

図 6　エージェントの応答文

通知

Windows の通知のストック領域にアクセスして通知データを取得する。各通知データに対して一度だけ、本システムの UI に表示する。

Web スクレイピング

定期的に twitter のトレンドと天気予報を取得する。

RSS フィード

設定した RSS フィードから定期的にランダムに記事を取得する。

5　評価

いくつかの入力に対して、エージェントがタスクに必要な情報を正しく解釈し、操作を実行できたことを確認した。また、ユーザの入力や常駐機能により取得したデータをに対して個性を持った応答文を生成できたことを確認した。

今回の実装では、語句や画像が存在しない空間をクリックするようなタスクを設定することができなかった。また、画面内に同じ語句が複数ある場合、最初に認識する語句しかクリックできなかった。

Tesseract を用いた OCR では、英語に対しては良好な結果を得たが、日本語のテキストに対しては正しい単語の区切りを認識することができていなかった。これは日本語が分かち書きされない言語であることやデスクトップ画面の語句の位置や大きさが不規則であるためと考える。

6　まとめ

本研究では、ユーザとエージェント間の自然言語による対話を通じてユーザの望むデスクトップ GUI の操作を実現するデスクトップ・エージェント・システムを開発した。

本システムは、UI、自然言語から DSL の変換、自動操作機能、応答文の生成、常駐機能の 5 つで構成される。UI には、GUI に加えて音声 UI を実装し、より実際のコミュニケーションと近いインタラクションを実現した。自然言語から DSL への変換は、GPT-3.5 の Function calling を利用することで、自然言語の命令から適切なタスクを認識することができた。加えて、DSL を定義することで、様々なタスクを実行可能とした。自動操作機能では、6 つの基本的な操作を実装し DSL で記述された操作を実行可能とした。デスクトップ画面の画像を解析することでマウス操作を自動化した。応答文の生成には、GPT-3.5 と記憶機能を実装し、キャラクター性を持ったエージェントの応答文を生成できた。また、常駐機能を実装することにより、ユーザがアクションを行わない間はサービ

スを提供できない点を改善し、より有用な体験の提供を可能とした。

7 今後の課題

本研究の課題として、以下のものが挙げられる。

- 現状のシステムでは画面内に同じ語句が複数ある場合、どの語句を選択するか判断できないため、それらを選択できるような手法の検討を行う。
- 自然言語から DSL への変換では、単一の入力のみに対して推論を行っている。より柔軟に対応するために、記憶機能と連携して以前の対話を考慮するように改修する。
- より多くの常駐機能を持つことでよりユーザの要望に応えられるように機能の拡充を行う。

参 考 文 献

[1] OpenAI. Openai. https://openai.com/.

[2] OpenAI. Gpt-4 technical report. https://arxiv.org/pdf/2303.08774.pdf, 2023.

[3] OpenAI. Language models are few-shot learners. https://arxiv.org/pdf/2005.14165.pdf, 2020.

[4] Ray Smith and Stefan Weil. Tesseract user manua. https://tesseract-ocr.github.io/tessdoc/\#introduction.

[5] Google. Google speech recognition. https://cloud.google.com/speech-to-text?hl=ja, 2011.

[6] Hiroshiba Kazuyuki. Voicevox — 無料のテキスト読み上げソフトウェア. https://voicevox.hiroshiba.jp/.

[7] Atty Eleti, Jeff Harris, and Logan Kilpatrick. Function calling and other api updates. https://openai.com/blog/function-calling-and-other-api-updates, year = 2023.

[8] OpenAI. Embeddings - openai api. https://platform.openai.com/docs/guides/embeddings.