2023年8月29日

教育システム情報学会全国大会プレコンファレンス

ChatGPTの自然言語処理エンジンとしての活用提案

慶應義塾大学/Bridge UI, Inc.　佐々木 雄司

Bridge UI 株式会社

概要

- アプリケーションの処理機構としてGPTを活用する
- GPT APIの使い方
- Function Calling

対象

・プログラミングの経験者
・教育のためのコンピュータシステムの開発・研究者
・すでにGPTを使ったことがある人

自由記述データ

Chat GPTを活用したい

AIは危険だと思う

説明の意味がわからなかった

CNNとはどう違うのか？

数値化

- 感情: Positive/Negative
- 危険性への言及: Boolean
- 理解度: スコア
- 分類: 度数分布

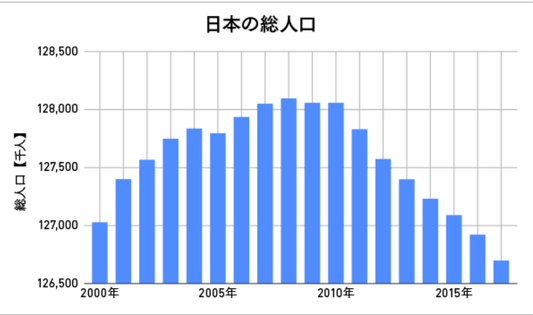
Bridge U

事例1

デジタル教科書/問題集

フィードバック

Q　このグラフから読み取れることは？



- 正誤判定
- 点数化
- 助言

人口が増え続けている　送信

不正解 / 2007年ごろまでとそれ以降で分けて考えましょう

Bridge UI

事例2

自己紹介

佐々木雄司

Bridge UI, Inc. 社長 兼 CTO

慶應義塾大学 政策メディア研究科

専門

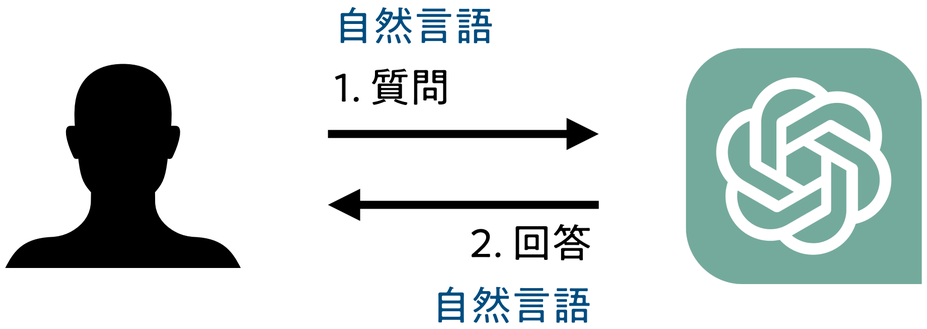
- Human-Computer Interaction
- 教育工学

y.sasaki@keio.jp

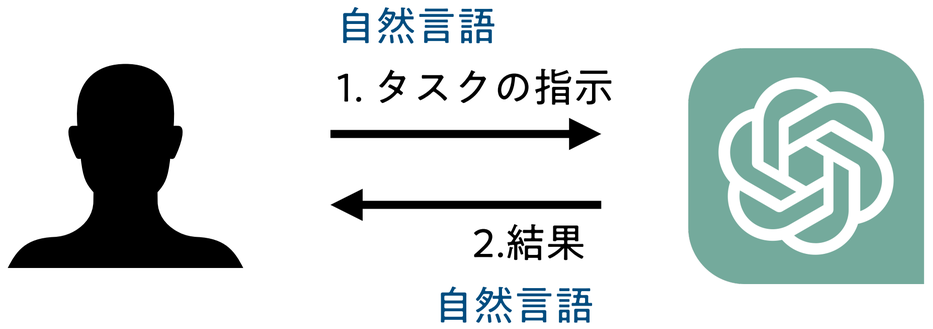
https://sasaki.dev

APIからChatGPTを使う

- 質問



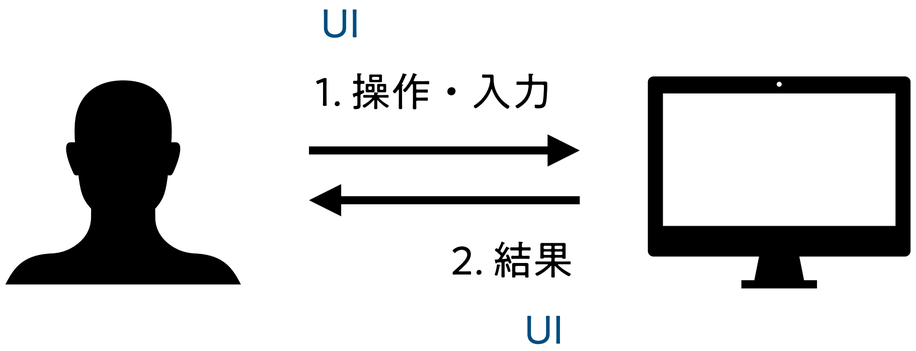
- 作業



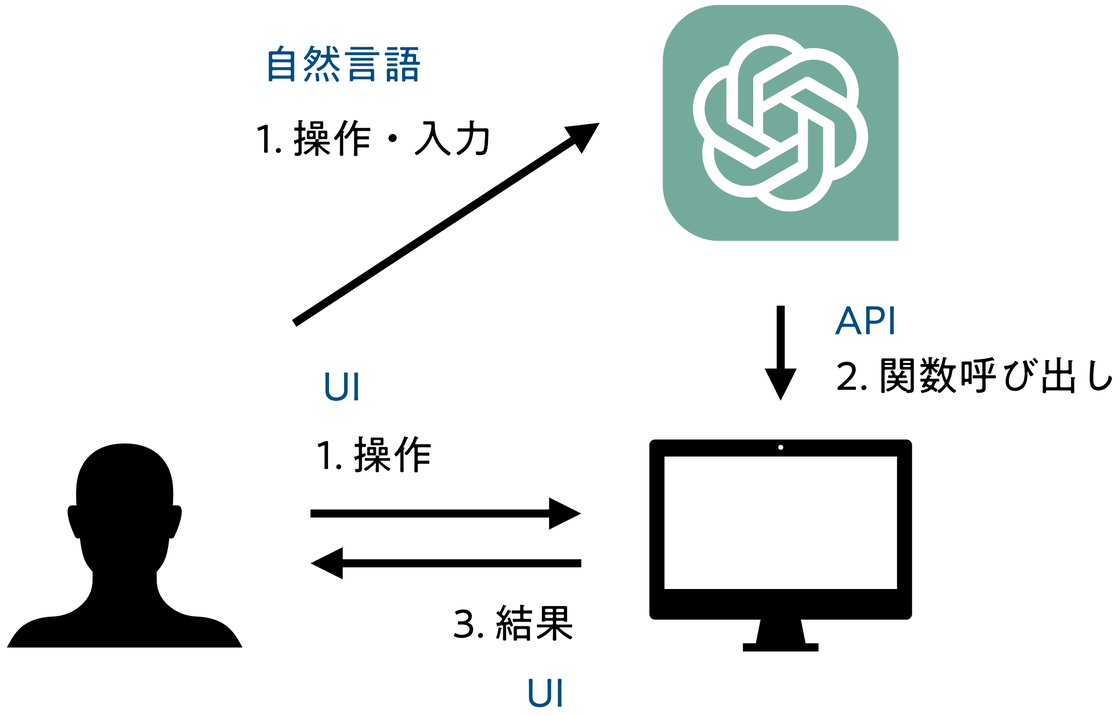
Bridge AI

会話で使う

- 従来



- w/ GPT



Bridge UI

システムのエンジンとして使う

- リアルタイムのフィードバック
- 生徒の回答・思考のデータ化

Bridge UI

教育系システムの要件

- リアルタイムのフィードバック
- 生徒の回答・思考のデータ化

⇩

従来：

・評価しやすい形式で回答させる（選択式・特殊なUI）

・生の記述を評価できない（自然言語処理が必要）

Bridge UI

教育系システムの要件

従来：

・評価しやすい形式で回答させる（選択式・特殊なUI）

・生の記述を評価できない（自然言語処理が必要）

⇩

GPT

・自由記述を評価（従来の自然言語処理を一部置き換え）

・直接的な評価が可能に

Bridge UI

教育系システムの要件

Chat GPT APIとは

仕様

- Open API（ChatGPTの開発元）が提供
  - ほかにもGPT APIはMicrosoftやGoogleなどが提供
  - モデル自体の配布も存在

- WebAPI

```
curl https://api.openai.com/v1/chat/completions \
 -H "Authorization: Bearer $OPENAI_API_KEY" \
 -H "Content-Type: application/json" \
 -d '{
 "model": "gpt-3.5-turbo",
 "messages": [{"role": "user", "content": "What is the OpenAI mission?"}]
 }'
```

仕様

- ライブラリの提供
  - 公式 Python/Node.js,
  - Azure .Net/JavaScript/Java/Go
  - コミュニティ 主要な様々な言語
- 様々なモデル（後述）
- fine-tuningも可能（後述）

様々なモデル

- モデルによって価格が異なる

- GPT-4 / GPT-3.5
  自然言語を理解して回答のテキストを生成

- DALL·E
  自然言語から画像を生成

- Whisper
  音声をテキストに変換

料金

- tokenごとに課金
- 1000 token ≒ 750 words（日本語の場合は英文より3倍割高と言われる）

例：こんにちは，世界！　→　こんにちは / , / 世界 / !

料金

Model	Input	Output
GPT-4 8K Context	$0.03 / 1K tokens	$0.06 / 1K tokens
GPT-4 32K Context	$0.06 / 1K tokens	$0.12 / 1K tokens
GPT-3.5 Turbo 4K Context	$0.0015 / 1K tokens	$0.002 / 1K tokens
GPT-3.5 Turbo 16K Context	$0.003 / 1K tokens	$0.004 / 1K tokens

fine-tuning

・モデルに追加で学習させて特定のユースケースに最適化させる手法

・会話の例を10件以上提示（50-100件で明確な改善）

・料金　GPT-3.5 < GPT-3.5 fine-tuned (8倍) < GPT4

・デメリット　料金・モデルが不安定になる危険・機能が少ない

https://platform.openai.com/docs/guides/fine-tuning

- 向かないタスク（シンプルな計算等）
  - あくまでも近似
  - 個数を数える，計算する，などのタスクはプログラミングが得意
- プライバシー
  - 許可しなければ学習には使われない（2023/3以降）
  - OpenAIの社員が見る可能性は否定できない
  - Azure OpenAI Service
- プロンプトインジェクション

Bridge U

利用上の注意

基本的な使い方

事前準備

- API Keyの取得
  https://platform.openai.com/account/api-keys
- ライブラリのインストール

```
pip install openai
```

ex1.py

```python
import openai

# API Keyを設定
openai.api_key = "#####"

chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[{"role": "user", "content": "Hello world"}]
    )

print(chat_completion)
```

結果


```json
{
  "id": "chatcmpl-7s7Cx16c65CkynGwrWYJln1EhZtNi",
  "object": "chat.completion",
  "created": 1693131775,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! How can I assist you today?"
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 9,
    "completion_tokens": 9,
    "total_tokens": 18
  }
}
```


Bridge AI

Hello World

ポイント

- API KeyとModel，Messageを設定
- Messageはroleとcontentを含むオブジェクトの配列
- role user→ユーザからの入力

ex2.py

```python
import openai

# API Keyを設定
openai.api_key = "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"

chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "入力された文章のタイトルをつける"},
        {"role": "user", "content": "ChatGPTを自然言語処理エンジンとして見ると，自由記述解答など従来コンピュータで処理が難しかったデータを簡単に扱える可能性が拡がる．しかし，通常のチャット形式での利用時には，応答が毎回異なるため一貫性を持たせたプログラミングが困難である．そこで，本講演ではプログラムからChatGPTを利用するAPIと，それをデータ分析やソフトウェアのバックグラウンドシステムにどのように組み込むかについて具体的な実装例とともに紹介する．"}
        ]
    )

print(chat_completion["choices"][0]["message"]["content"])
```

結果

- 「ChatGPTを活用した自然言語処理の可能性と具体的な実装例について紹介します」

何かをしてくれるシステム

ポイント

- role system→何をするシステムなのか
- role user→ユーザからの入力

何かをしてくれるシステム

ex3-1.py

```python
import openai

# API Keyを設定
openai.api_key = "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"

chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "しりとりをしてください"},
        {"role": "user", "content": "りんご"}
        ]
    )

print(chat_completion["choices"][0]["message"]["content"])
```

結果

- 「ごま」

ポイント

- GPTは状態を持つことができない
- 過去の会話を全て含めて次の回答を要求する必要がある
- 過去のGPTからの回答はrole assistantで表す

ex3-2.py

```python
import openai

# API Keyを設定
openai.api_key = "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"

chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "しりとりをしてください"},
        {"role": "user", "content": "りんご"},
        {"role": "assistant", "content": "ごま"},
        {"role": "user", "content": "まき"},
        ]
    )

print(chat_completion["choices"][0]["message"]["content"])
```

結果

- 「きつね」

Function Calling

概要

- 外部関数を使って回答できるようにする仕組み
- 例：天気取得APIで「今の東京の天気は？」に答える
- レスポンスを自然言語ではなくコンピュータが理解できる形式にするHackに使える
    - 通常のレスポンスは揺らぎがあるのでプログラムから扱いづらい

ex4-1.py

```python
import openai

# API Keyを設定
openai.api_key = "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"

def get_temperture_function(location: str, unit: str = "cercius"):
    # 天気を取得するコード
    return 30

messages = [{"role": "user", "content": "大阪の気温は？"}]
chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=messages,
    functions=[
        {
            "name": "get_temp",
            "description": "与えられた位置の情報から気温を返す",
            "parameters": {
                "type": "object",
                "properties": {
                    "location": {
                        "type": "string",
                        "description": "都市 e.g.東京",
                    },
                    "unit": {"type": "string", "enum": ["celsius", "fahrenheit"]},
                },
                "required": ["location"],
            }
        }
    ]
    )

return_message = chat_completion["choices"][0]["message"]
print(return_message)

```

結果

```
{
  "role": "assistant",
  "content": null,
  "function_call": {
    "name": "get_temp",
    "arguments": "{\n  \"location\": \"\u5927\u962a\"\n}"
  }
}
```

基本的なFunction Calling

ex4-2.py

```
1
2 return_message = chat_completion["choices"][0]["message"]
3 if return_message.get("function_call"):
4     function_list = { "get_temp": get_temperture_function }
5     function_to_call = function_list[return_message["function_call"]["name"]]
6     function_args = json.loads(return_message["function_call"]["arguments"])
7     function_response = function_to_call(
8         location=function_args.get("location"),
9         unit=function_args.get("unit"),
10    )
11
12    messages.append(return_message)
13    messages.append(
14                {
15            "role": "function",
16            "name": return_message["function_call"]["name"],
17            "content": function_response,
18        }
19    )
20
21    second_response = openai.ChatCompletion.create(
22        model="gpt-3.5-turbo",
23        messages=messages,
24    )
25
26    print(second_response["choices"][0]["message"]["content"])
27
```

結果

- 「大阪の現在の気温は30℃です。

基本的なFunction Calling

ユースケース

- 生徒の回答をリアルタイムで評価・フィードバックする教材を開発したい
- 「掛け算を使った問題をつくりなさい」という問題への回答を評価して正誤とフィードバックを返す

ex5.py

```python
import openai
import json

# API Keyを設定
openai.api_key = "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"

chat_completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo-0613",
    messages=[
        { "role": "system", "content": "「掛け算を使った問題をつくりなさい」という問題への回答が入力されます．入力が問題への回答として正しいかどうかを評価して，正誤とフィードバック文を表示せよ"},
        { "role": "user", "content": "太郎が花を3本，次郎が花を10本もっています．合わせていくつ？"}
    ],
    functions=[
        {
            "name": "show",
            "description": "正誤とフィードバック文を表示",
            "parameters": {
                "type": "object",
                "properties": {
                    "reason": {
                        "type": "string",
                        "description": "間違っている理由 e.g. これは割り算の問題です",
                    },
                    "result": {"type": "string", "enum": ["correct", "incorrect"]},
                },
                "required": ["result"],
            }
        }
    ]
    )
print(chat_completion["choices"][0]["message"]["function_call"]["arguments"])
```

結果

```
{
  "reason": "これは足し算の問題です",
  "result": "incorrect"
}
```

Function Callingのjson出力利用

プロンプトエンジニアリング

- Write clear instructions　明確な指示を書く
- Provide reference text　参照テキストの提供
- Split complex tasks into simpler subtasks　サブタスクへの分割
- Give GPTs time to “think”　考える時間を与える（推論ステップを増やす）
- Use external tools　外部ツールの使用
- Test changes systematically　変更を体系的にテスト

Bridge U

良い結果を得るための6つの戦略

ルーブリック評価の活用

例：レポートの評価

「このレポートを100点満点で採点してください」

↓

評価があまりにもブラックボックスになってしまう

ルーブリックによる評価

「適切なテーマ設定をしているかについて評価します．課題に応じた適切な…されていれば4点，…」

評価の観点	観点の説明	評価の基準			
		4	3	2	1
テーマ設定	適切なテーマ設定をしているか。	課題に応じた適切なテーマ設定が簡潔かつ明確になされている。	課題に応じたテーマ設定がなされている。	課題に応じたテーマ設定がなされているが、工夫の余地がある。	課題に応じたテーマが設定されていない。
クリティカル・シンキング	批判的考察がなされているか（クリティカル・シンキングの観点）。	先行研究や事例、データなどを基に、批判的考察が明確かつ十分になされている。	先行研究や事例、データなどを基に、批判的考察がなされている。	批判的考察はなされているが、先行研究や事例、データなどに基づいていない。	批判的考察がなされていない。
根拠の成立	主張（考察）するための根拠（データや先行研究等）が示され、分析できているか。	主張したい論点に対する根拠が明確かつ適切に示されて、高度で客観的な分析ができている。	主張したい論点に対する根拠が適切に示され、客観的な分析ができている。	主張したい論点に対する根拠が示されているが、客観的な分析ができていない。	主張したい論点に対する考察が示されておらず、客観的な分析ができていない。
論理的な構成	論理的に構成されているか（ロジカル・シンキングの観点）。	主張したい論点に対して、構造的に整理されており、一貫性と納得性（説明力）のある構成となっている。	主張したい論点に対して、構造的に整理されているが、一貫性や納得性（説明力）の点で改善の余地がある。	主張したい論点に対して、論理的に構成しようとしていることは認められる。	主張したい論点に対して、論理的に構成されていない。

龍谷大学「ルーブリック作成ガイドブック」https://fd.ryukoku.ac.jp/biz_content2/project1/data/rubric.pdf

Bridge U

良い結果を得るための6つの戦略

生徒が書いた文章・回答の評価

- Write clear instructions → ルーブリックの提示
- Provide reference text → 専門用語などの参考文献を与える
- Split complex tasks into simpler subtasks → 評価項目毎にAPIを切り分け
- Give GPTs time to “think” → 1つの評価項目の推論ステップを複数に
- Use external tools → 数学計算などにはプラグイン / function callingを使用
- Test changes systematically

Bridge U

良い結果を得るための6つの戦略

まとめ

- ChatGPTのAPIを使うと生のデータを活用しやすくなる
  - 教材開発で自由記述をリアルタイム評価
  - アンケートの自由記述の分析
- Function Callingを使えば安定してプログラムからChatGPTを使用できる
- 自由記述の分析にはルーブリックの活用が有効

デジタル庁「ChatGPTを業務に組み込むためのハンズオン」（https://www.digital.go.jp/assets/contents/node/information/field_ref_resources/5896883b-cc5a-4c5a-b610-eb32b0f4c175/82ccd074/20230725_resources_ai_outline.pdf）2023/08/27参照

OpenAI「Introducing ChatGPT and Whisper APIs」（https://openai.com/blog/introducing-chatgpt-and-whisper-apis）2023/08/27参照

OpenAI「API Reference - Open API」（https://platform.openai.com/overview）2023/08/27参照

龍谷大学「ルーブリック作成ガイドブック」（https://fd.ryukoku.ac.jp/biz_content2/project1/data/rubric.pdf）2023/08/27参照

例示したプログラム

https://github.com/jyu0414/chatgpt-api-example

参考資料

2023年8月29日

教育システム情報学会全国大会プレコンファレンス

ChatGPTの自然言語処理エンジンとしての活用提案

慶應義塾大学/Bridge UI, Inc.　佐々木 雄司

Bridge UI 株式会社