ChatGPT のプログラム

金子邦彦

1. 基本的な説明
2. 次のページのプログラム実行結果（プログラムを一部変更して使用している）

   https://note.com/npaka/n/n0fd7bd3ed27b

ChatGPT の最新ニュース 2023/11/7

次の予定がアナウンスされた

- ChatGPT-4 Turbo　トークン数 128,000個
- 値下げ
- 視覚、聴覚との統合
- GPTs 利用者が、独自のデータや指示を追加することにより、独自の ChatGPT を作成可能

テキスト生成モデルとチャットモデル

テキスト生成モデル

- **与えられた入力テキストに基づき、新たなテキストを生成**
- 記事の作成、コード生成、文の創作、質問に対する応答、要約の作成など
- テキストを自動的に生成。多岐にわたる生成が可能

チャットモデル

- **人間との対話を再現することを目的としたテキスト生成モデルの一種**
- 対話の流れを把握し、受けた質問に応答。
- 顧客サポート、対話型アシスタントなど、リアルタイムでの対話が求められる環境を目的とする

テキスト生成モデルとチャットモデル

テキスト生成モデル

- 多目的に利用され、大規模なテキストの生成が可能
- 柔軟性により幅広い応用が可能

チャットモデル

- ユーザーとの対話を重視
- 過去の対話に沿った応答を生成。一貫性のある会話の流れを実現

ChatGPT

- チャットモデルの一種。テキスト生成モデルの能力も持つ
- 対話形式のインタラクションに特化。会話を通じて質問に答えることを目的。対話から学び、より適切な応答を生成する能力も持つ。

OpenAI の APIキー

- OpenAIが提供する様々な人工知能モデルにアクセスするための API キー
- 無料でも、有料でも利用可能
- オンラインで取得

準備①

- **OpenAI の APIキーを取得**

OpenAI の APIキーのページ

https://platform.openai.com/account/api-keys

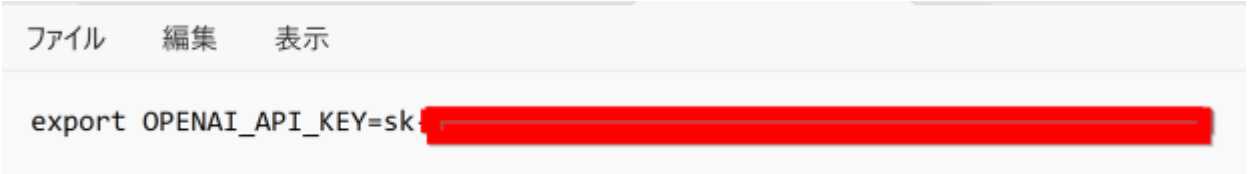
- **エディタを起動。次のコマンドを実行**

```
cd %HOMEPATH%
rmdir /s /q langchain
mkdir langchain
cd langchain
type nul > .env
notepad .env
```

準備②

- **エディタで 「export OPENAI_API_KEY=sk-...」のように書く.**

「sk-...」の部分は，OpenAI の APIキー を設定する.



以上で %HOMEPATH%¥langchain¥.env に OPENAI キーが設定できた

1．テキスト生成モデル (OpenAI キー必要)

- **前準備**: コマンドプロンプトを管理者として実行．次のコマンドを実行

pip install -U langchain==0.0.329 openai==v0.28.1

cd %HOMEPATH%

cd langchain

1．テキスト生成モデル (OpenAI キー必要)

Python プログラム

```
import dotenv
dotenv.load_dotenv()
from langchain.llms import OpenAI
llm = OpenAI(temperature=0.9)
print(llm.predict("卒業論文の書き方をアドバイスしてください。"))
```

```
C:¥Users¥user¥langchain>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dotenv
>>> dotenv.load_dotenv()
True
>>> from langchain.llms import OpenAI
>>> llm = OpenAI(temperature=0.9)
>>> print(llm.predict("卒業論文の書き方をアドバイスしてください。"))


1.まずは、論文のテーマを明確に決めることから始めましょう。それは論文内容を定義するためです。

2.研究を行う前に、学術文献や関連情報を調査して、専門分野で展開することを学びましょう。

3.論文の枠組みを確立しましょう。次に、仮説を検証して、結論を導き出すために、アーカイブ、データ収集、統計分析などを行う必要がある
>>>
```

```
C:¥Users¥user¥langchain>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dotenv
>>> dotenv.load_dotenv()
True
>>> from langchain.llms import OpenAI
>>> llm = OpenAI(temperature=0.9)
>>> print(llm.predict("卒業論文の書き方をアドバイスしてください。"))


1.まずは、論文のテーマを明確に決めることから始めましょう。それは論文内容を定義するためです。

2.研究を行う前に、学術文献や関連情報を調査して、専門分野で展開することを学びましょう。

3.論文の枠組みを確立しましょう。次に、仮説を検証して、結論を導き出すために、アーカイブ、データ収集、統計分析などを行う必要がある
>>>
```

2．チャットモデル (OpenAI キー必要)

- **前準備**: コマンドプロンプトを管理者として実行．次のコマンドを実行　（１と同じ）

pip install -U langchain==0.0.329 openai==v0.28.1

cd %HOMEPATH%

cd langchain

2．チャットモデル (OpenAI キー必要)

Python プログラム

```
import dotenv

dotenv.load_dotenv()

from langchain.chat_models import ChatOpenAI

from langchain.schema import HumanMessage

chat_model = ChatOpenAI(temperature=0.9)

messages = [HumanMessage(content="卒業論文の書き方をアドバイスしてください。")]

print(chat_model.predict_messages(messages))
```

```
C:¥Users¥user¥langchain>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dotenv
>>> dotenv.load_dotenv()
True
>>> from langchain.chat_models import ChatOpenAI
>>> from langchain.schema import HumanMessage
>>> chat_model = ChatOpenAI(temperature=0.9)
>>> messages = [HumanMessage(content="卒業論文の書き方をアドバイスしてください。")]
>>> print(chat_model.predict_messages(messages))
content='卒業論文の書き方には様々なアプローチがありますが、以下のアドバイスを参考にしてください。¥n¥n1. テーマの選択：興味のあるテーマを選び、適切な範囲を設定してください。テーマが広すぎると調査や分析が困難になるため注意が必要です。¥n¥n2. 目的と仮説の設定：論文の目的や仮説を明確にしましょう。これにより、論文の方向性を定めることができます。¥n¥n3. 文献調査：関連する文献を網羅的に調査し、適切な参考文献を集めましょう。学術的な信頼性の高い文献を選ぶことが重要です。¥n¥n4. データの収集：実証的な論文の場合、自身でデータを収集する必要があるかもしれません。調査手法やデータ収集方法を明記し、信頼性や妥当性を考慮してください。¥n¥n5. 分析方法：データを分析するために適切な統計手法や分析手法を選びましょう。適切な手法を選ぶことで結果を客観的に評価することができます。¥n¥n6. 論文の構成：序論、方法、結果、考察、結論など、論文の構成を考えましょう。それぞれのセクションで適切な情報を提供し、論文の流れを明確にしましょう。¥n¥n7. 論文の執筆：明確で簡潔な文章を心掛け、構造化されたアウトラインに基づいて執筆しましょう。また、文献引用の規則に従い、引用文献リストを作成してください。¥n¥n8. レビューと編集：論文が完成したら、同僚や教員にレビューしてもらいましょう。論理的なフロー、誤字や文法のミス、主張の正確性などを確認し、必要であれば修正してください。¥n¥n9. フォーマットと提出：大学の指示やガイドラインに従い、適切なフォーマットで論文を仕上げましょう。提出前に最終的なチェックを行い、不備やミスがないことを確認してください。¥n¥n以上のアドバイスは一般的なものですが、大学や学部によって要件や指示が異なる場合があります。そのため、指導教員や学部のガイドラインに従うことも重要です。'
>>>
```

```
C:¥Users¥user¥langchain>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dotenv
>>> dotenv.load_dotenv()
True
>>> from langchain.chat_models import ChatOpenAI
>>> from langchain.schema import HumanMessage
>>> chat_model = ChatOpenAI(temperature=0.9)
>>> messages = [HumanMessage(content="卒業論文の書き方をアドバイスしてください。")]
>>> print(chat_model.predict_messages(messages))
content='卒業論文の書き方には様々なアプローチがありますが、以下のアドバイスを参考にしてください。¥n¥n1. テーマの選択：興味のあるテーマを選び、適切な範囲を設定してください。テーマが広すぎると調査や分析が困難になるため注意が必要です。¥n¥n2. 目的と仮説の設定：論文の目的や仮説を明確にしましょう。これにより、論文の方向性を定めることができます。¥n¥n3. 文献調査：関連する文献を網羅的に調査し、適切な参考文献を集めましょう。学術的な信頼性の高い文献を選ぶことが重要です。¥n¥n4. データの収集：実証的な論文の場合、自身でデータを収集する必要があるかもしれません。調査手法やデータ収集方法を明記し、信頼性や妥当性を考慮してください。¥n¥n5. 分析方法：データを分析するために適切な統計手法や分析手法を選びましょう。適切な手法を選ぶことで結果を客観的に評価することができます。¥n¥n6. 論文の構成：序論、方法、結果、考察、結論など、論文の構成を考えましょう。それぞれのセクションで適切な情報を提供し、論文の流れを明確にしましょう。¥n¥n7. 論文の執筆：明確で簡潔な文章を心掛け、構造化されたアウトラインに基づいて執筆しましょう。また、文献引用の規則に従い、引用文献リストを作成してください。¥n¥n8. レビューと編集：論文が完成したら、同僚や教員にレビューしてもらいましょう。論理的なフロー、誤字や文法のミス、主張の正確性などを確認し、必要であれば修正してください。¥n¥n9. フォーマットと提出：大学の指示やガイドラインに従い、適切なフォーマットで論文を仕上げましょう。提出前に最終的なチェックを行い、不備やミスがないことを確認してください。¥n¥n以上のアドバイスは一般的なものですが、大学や学部によって要件や指示が異なる場合があります。そのため、指導教員や学部のガイドラインに従うことも重要です。'
>>>
```

３．メモリを用いたチャットモデル(OpenAI キー必要)

いままでの会話をメモリに記憶し、その記憶も使用して、返答を生成する

- **前準備**: コマンドプロンプトを管理者として実行．次のコマンドを実行　（１と同じ）

pip install -U langchain==0.0.329 openai==v0.28.1

cd %HOMEPATH%

cd langchain

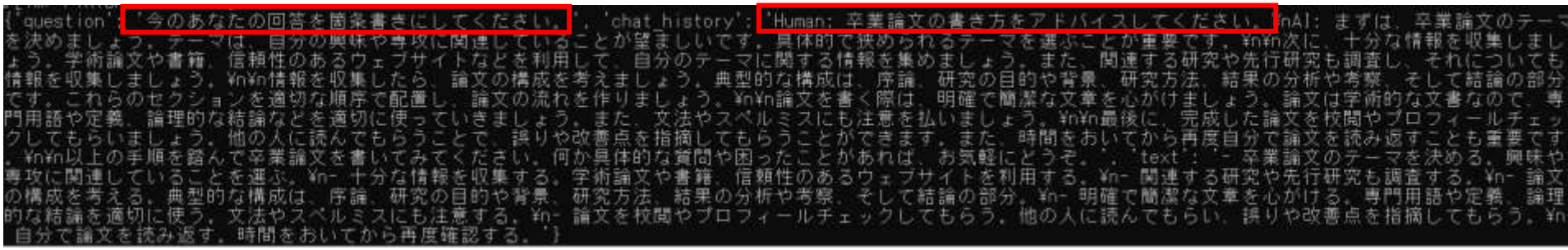
３．メモリを用いたチャットモデル (OpenAI キー必要)

Python プログラム

```
import dotenv
dotenv.load_dotenv()
from langchain.chains import LLMChain
from langchain.chat_models import ChatOpenAI
from langchain.memory import ConversationBufferMemory
from langchain.prompts import PromptTemplate
chat_model = ChatOpenAI(temperature=0.9)

# PromptTemplateの準備
template = """You are a helpful assistant. Let's take a deep breath and tackle this issue one step at a time.

Previous conversation:
{chat_history}
New human question: {question}
Response:"""
prompt = PromptTemplate.from_template(template)
memory = ConversationBufferMemory(memory_key="chat_history")
conversation = LLMChain(
llm=chat_model,
prompt=prompt,
memory=memory,
verbose=True,
)
conversation({"question": "卒業論文の書き方をアドバイスしてください。"})
conversation({"question": "今のあなたの回答を箇条書きにしてください。"})
```

```
C:¥Users¥user¥langchain>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dotenv
>>> dotenv.load_dotenv()
True
>>> from langchain.chains import LLMChain
>>> from langchain.chat_models import ChatOpenAI
>>> from langchain.memory import ConversationBufferMemory
>>> from langchain.prompts import PromptTemplate
>>> chat_model = ChatOpenAI(temperature=0.9)
>>>
>>> # PromptTemplateの準備
>>> template = """You are a helpful assistant. Let's take a deep breath and tackle this issue one step at a time.
...
... Previous conversation:
... {chat_history}
... New human question: {question}
... Response:"""
>>> prompt = PromptTemplate.from_template(template)
>>> memory = ConversationBufferMemory(memory_key="chat_history")
>>> conversation = LLMChain(
...     llm=chat_model,
...     prompt=prompt,
...     memory=memory,
...     verbose=True,
... )
>>> conversation({"question": "卒業論文の書き方をアドバイスしてください。"})


[1m> Entering new LLMChain chain...[0m
Prompt after formatting:
[32;1m[1;3mYou are a helpful assistant. Let's take a deep breath and tackle this issue one step at a time.

Previous conversation:

New human question: 卒業論文の書き方をアドバイスしてください。
Response:[0m

[1m> Finished chain.[0m
{'question': '卒業論文の書き方をアドバイスしてください。', 'chat_history': '', 'text': 'まずは、卒業論文のテーマを決めましょう。テーマは、自分の興味や専攻に関連していることが望ましいです。具体的で狭められるテーマを選ぶことが重要です。¥n¥n次に、十分な情報を収集しましょう。学術論文や書籍、信頼性のあるウェブサイトなどを利用して、自分のテーマに関する情報を集めましょう。また、関連する研究や先行研究も調査し、それについても情報を収集しましょう。¥n¥n情報を収集したら、論文の構成を考えましょう。典型的な構成は、序論、研究の目的や背景、研究方法、結果の分析や考察、そして結論の部分です。これらのセクションを適切な順序で配置し、論文の流れを作りましょう。¥n¥n論文を書く際は、明確で簡潔な文章を心がけましょう。論文は学術的な文書なので、専門用語や定義、論理的な結論などを適切に使っていきましょう。また、文法やスペルミスにも注意を払いましょう。¥n¥n最後に、完成した論文を校閲やプロフィールチェックしてもらいましょう。他の人に読んでもらうことで、誤りや改善点を指摘してもらうことができます。また、時間をおいてから再度自分で論文を読み返すことも重要です。¥n¥n以上の手順を踏んで卒業論文を書いてみてください。何か具体的な質問や困ったことがあれば、お気軽にどうぞ。'}
>>> conversation({"question": "今のあなたの回答を箇条書きにしてください。"})


[1m> Entering new LLMChain chain...[0m
Prompt after formatting:
[32;1m[1;3mYou are a helpful assistant. Let's take a deep breath and tackle this issue one step at a time.

Previous conversation:
Human: 卒業論文の書き方をアドバイスしてください。
AI: まずは、卒業論文のテーマを決めましょう。テーマは、自分の興味や専攻に関連していることが望ましいです。具体的で狭められるテーマを選ぶことが重要です。

次に、十分な情報を収集しましょう。学術論文や書籍、信頼性のあるウェブサイトなどを利用して、自分のテーマに関する情報を集めましょう。また、関連する研究や先行研究も調査し、それについても情報を収集しましょう。

情報を収集したら、論文の構成を考えましょう。典型的な構成は、序論、研究の目的や背景、研究方法、結果の分析や考察、そして結論の部分です。これらのセクションを適切な順序で配置し、論文の流れを作りましょう。

論文を書く際は、明確で簡潔な文章を心がけましょう。論文は学術的な文書なので、専門用語や定義、論理的な結論などを適切に使っていきましょう。また、文法やスペルミスにも注意を払いましょう。

最後に、完成した論文を校閲やプロフィールチェックしてもらいましょう。他の人に読んでもらうことで、誤りや改善点を指摘してもらうことができます。また、時間をおいてから再度自分で論文を読み返すことも重要です。

以上の手順を踏んで卒業論文を書いてみてください。何か具体的な質問や困ったことがあれば、お気軽にどうぞ。
New human question: 今のあなたの回答を箇条書きにしてください。
Response:[0m

[1m> Finished chain.[0m
{'question': '今のあなたの回答を箇条書きにしてください。', 'chat_history': 'Human: 卒業論文の書き方をアドバイスしてください。¥nAI: まずは、卒業論文のテーマを決めましょう。テーマは、自分の興味や専攻に関連していることが望ましいです。具体的で狭められるテーマを選ぶことが重要です。¥n¥n次に、十分な情報を収集しましょう。学術論文や書籍、信頼性のあるウェブサイトなどを利用して、自分のテーマに関する情報を集めましょう。また、関連する研究や先行研究も調査し、それについても情報を収集しましょう。¥n¥n情報を収集したら、論文の構成を考えましょう。典型的な構成は、序論、研究の目的や背景、研究方法、結果の分析や考察、そして結論の部分です。これらのセクションを適切な順序で配置し、論文の流れを作りましょう。¥n¥n論文を書く際は、明確で簡潔な文章を心がけましょう。論文は学術的な文書なので、専門用語や定義、論理的な結論などを適切に使っていきましょう。また、文法やスペルミスにも注意を払いましょう。¥n¥n最後に、完成した論文を校閲やプロフィールチェックしてもらいましょう。他の人に読んでもらうことで、誤りや改善点を指摘してもらうことができます。また、時間をおいてから再度自分で論文を読み返すことも重要です。¥n¥n以上の手順を踏んで卒業論文を書いてみてください。何か具体的な質問や困ったことがあれば、お気軽にどうぞ。', 'text': '1. 卒業論文のテーマを決める：興味や専攻に関連していることを選ぶ。¥n2. 十分な情報を収集する：学術論文や書籍、信頼性のあるウェブサイトを利用する。¥n3. 関連する研究や先行研究も調査する。¥n4. 論文の構成を考える：典型的な構成は、序論、研究の目的や背景、研究方法、結果の分析や考察、そして結論の部分。¥n5. 明確で簡潔な文章を心がける：専門用語や定義、論理的な結論などを適切に使う。文法やスペルミスにも注意する。¥n6. 論文を校閲やプロフィールチェックしてもらう：他の人に読んでもらい、誤りや改善点を指摘してもらう。¥n7. 自分で論文を読み返す：時間をおいてから再度確認する。'}
>>>
```

最新のプロンプト　　過去のプロンプト



現在までの回答すべて

４．検索拡張生成 (OpenAI キー必要）

「検索拡張生成」(RAG:Retrieval Augmented Generation)
外部データを使用。生成ステップの実行時にLLMに追加情報として渡す。

- **前準備**: コマンドプロンプトを管理者として実行．次のコマンドを実行．data¥doc.txt は外部データのファイル

```
pip install -U langchain==0.0.329 openai==v0.28.1
pip install chromadb tiktoken langchainhub unstructured
cd %HOMEPATH%
cd langchain
mkdir data
notepad data¥doc.txt
```

４．検索拡張生成(OpenAI キー必要）

Python プログラム

```
import dotenv
dotenv.load_dotenv()
from langchain.document_loaders import DirectoryLoader

# ドキュメントのロード
loader = DirectoryLoader('./data/')
documents = loader.load()
print(documents)
from langchain.text_splitter import CharacterTextSplitter
# ドキュメントの分割
text_splitter = CharacterTextSplitter(
chunk_size=1000,
chunk_overlap=20
)
splits = text_splitter.split_documents(documents)
# チャンクの確認
for i in range(len(splits)):
print(i, len(splits[i].page_content), splits[i].page_content)

from langchain.vectorstores import Chroma
from langchain.embeddings import OpenAIEmbeddings

# VectorStoreの準備
vectorstore = Chroma.from_documents(
documents=splits,
embedding=OpenAIEmbeddings()
)

retriever = vectorstore.as_retriever(search_kwargs={"k": 2})

from langchain.chat_models import ChatOpenAI
from langchain.schema.runnable import RunnablePassthrough
from langchain import hub

# LanguageModelの準備
llm = ChatOpenAI(
model_name="gpt-3.5-turbo",
temperature=0
)

# PromptTemplateの準備
rag_prompt = hub.pull("rlm/rag-prompt")

# RAGChainの準備
rag_chain = {
"context": retriever,
"question": RunnablePassthrough()
} | rag_prompt | llm

rag_chain.invoke("日本国憲法について説明")
```

```
>>> rag_chain.invoke("日本国憲法について説明")
AIMessage(content='日本国憲法は、日本国民が選挙で選ばれた国会を通じて行動し、国民の安全と平和を確保するために制定されたものであり、天皇は日本国の象徴であり、主権は国民に存するとされています。天皇の地位は、主権の存する日本国民の総意に基づいており、天皇の国事に関する行為には内閣の助言と承認が必要です。また、天皇は国政に関する権能を持たず、憲法で定められた国事に関する行為のみを行います。')
>>>
```

全体まとめ

テキスト生成モデルとチャットモデル

- テキスト生成モデル: 任意の入力から新たなテキストを生成
- チャットモデル: 人間との対話を目的とし、リアルタイムの応答を生成
- ChatGPT: チャットモデルに分類され、対話から学習して応答を改善

OpenAI APIキー

- AIモデルへのアクセスを管理するためのキー
- OpenAIのAPIキー取得ページで入手

プログラムの準備手順

- OpenAI APIキーを取得して環境変数に設定
- 必要なライブラリのインストール
- 環境設定ファイルの作成と編集

具体的なプログラムの例

- Python を用いて、テキスト生成モデルとチャットモデルを実行
- メモリを用いたチャットモデルで会話の履歴を利用
- 検索拡張生成（RAG）を利用して外部データを参照