AI 特許紹介(66)
AI 特許を学ぶ！究める！
～プログラムプロンプト特許～

2024 年 7 月 10 日
河野特許事務所
所長弁理士　河野英仁

「AI 特許紹介」シリーズは、注目すべき AI 特許のポイントを紹介します。熾烈な競争となっている第４次産業革命下では AI 技術がキーとなり、この AI 技術・ソリューションを特許として適切に権利化しておくことが重要であることは言うまでもありません。

AI 技術は Google, Microsoft, Amazon を始めとした IT プラットフォーマ、研究機関及び大学から毎週のように新たな手法が提案されており、また AI 技術を活用した新たなソリューションも次々とリリースされています。

本稿では米国先進 IT 企業を中心に、これらの企業から出願された AI 特許に記載された AI テクノロジー・ソリューションのポイントをわかりやすく解説致します。

1.概要
特許出願人　NVIDIA
出願日　2023 年 3 月 16 日
公開日　2024 年 3 月 21 日
公開番号 US20240095077
発明の名称　1 つ以上の機械学習プロセスで使用するプロンプトジェネレータ

077 特許は、プロンプト生成機能によりロボットを動作させるためのプロンプトを生成し、生成したプロンプトを大規模言語モデル(LLM)に与えることにより、ロボットがとるべき計画を生成するプログラムプロンプト技術に関する。

2.特許内容の説明
計画を立てるには、ロボットが行動する必要がある世界に関する無数のドメイン知識を定義することが必要になる。その労力を軽減するために、LLM を使用して、タスク計画中に潜在的な次のアクションにスコアを付けたり、追加のドメイン情報なしで自然言語の指示を与えてアクションシーケンスを直接生成したりすることもできる。

しかしながら、このような方法では、スコアリングのためにすべての可能な次のステップを列挙するか、現在のコンテキストで特定のロボットでは実行できないアクションを含む可能性のある自由形式のテキストを生成する必要がある。

077 特許では、状況に応じた環境、ロボットの機能、タスクにわたって機能的な計画生成を可能にするプログラム的なプロンプト構造を LLM に提示し、ロボットに対する計画を生成する。

図 1 は、システム 100 を示すブロック図を示す。

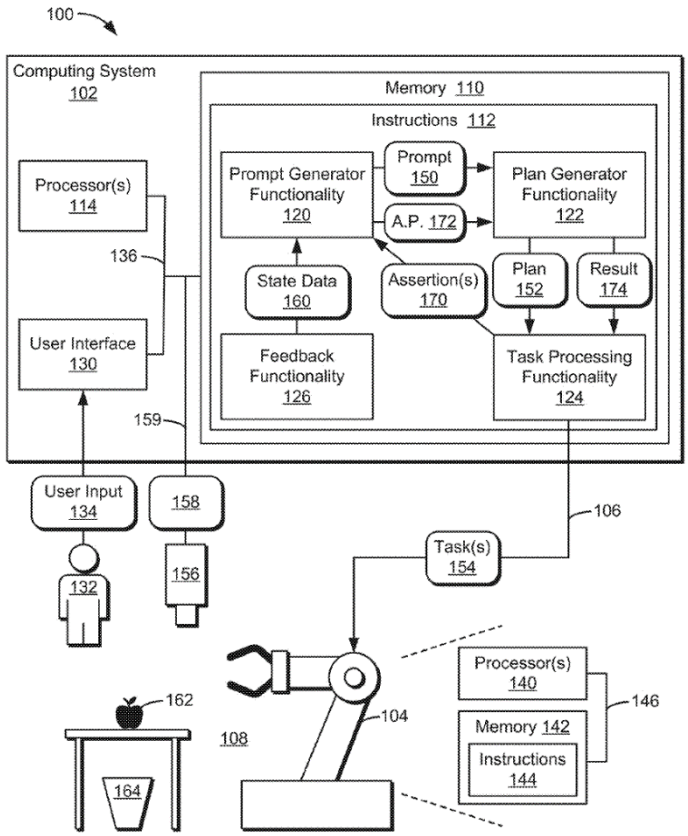


FIG. 1

システム 100 は、エージェント 104 と通信するコンピューティングシステム 102 を含む。エージェント 104 はロボットであり、仮想環境または現実世界環境である環境 108 内で動作する。コンピューティングシステム 102 は、プロンプト生成機能 120、計画生成機能 122、タスク処理機能 124、フィードバック機能 126 を実装するプロセッサ実行可能命令 112 を格納するメモリ 110 を含む。

プロンプト生成機能 120 は、プロンプト 150 を生成し、プロンプト 150 を計画生成機能 122 に提供する。プロンプト生成機能 120 は、ユーザ入力 134 でユーザ 132 から高レベルタスクの識別情報を受信する。高レベルタスクは、環境 108 を初期状態（たとえば、変数 s で表される）から目標状態（たとえば、変数 g で表される）に遷移させることを目指す。

計画生成機能 122 は、高レベルのタスクを、エージェント 104 が実行可能な低レベルのタスクに分割することができる。言い換えると、計画生成機能 122 は、プロンプト 150 を入力として受信し、エージェント 104 が実行可能な 1 つ以上のタスク 154 を含む計画 152 を出力する。

図 1 に示す例では、ユーザ 132 は（ユーザインターフェイス 130 を介して）プロンプト生成機能 120 への高レベルタスクを特定する。この例では、高レベルタスクは、エージェント 104 がリンゴ 162 を捨てることである。プロンプト生成機能 120 はプロンプト 150 を生成し、プロンプト 150 を計画生成機能 122 に送信する。計画生成機能 122 は計画 152 を生成する。

計画生成機能 122 は、コンピューティングシステム 102 またはエージェント 104 によって実装されるタスク処理機能 124 に計画 152 を転送する。計画 152 には、エージェント 104 によって実行されるタスク 154 が含まれる。計画 152 には、タスク 154 に加えて、コメントやアサーションなどの情報も含まれる。

アサーションは、エージェント 104 によって実行される将来のタスク（次のタスクなど）の前提条件となる環境 108 に関する何かをアサートする。たとえば、アサーションは、エージェント 104 がゴミ箱 164 の近くにあることをアサートする場合があり、これは、エージェント 104 がリンゴ 162 をゴミ箱 164 に入れるために必要とされる。

計画 152 にアサーションが含まれている場合、そのアサーションの真実性を判断する必要がある。コンピューティングシステム 102 およびエージェント 104 によって実装されたタスク処理機能 124 が計画 152 内の 1 つ以上のアサーション 170 に遭遇すると、タスク処理機能 124 は、アサーション 170（たとえば、各アサーションに遭遇するたびに）をプロンプト生成機能 120 に転送する。

プロンプト生成機能 120 は、アサーション 170 ごとにアサーションプロンプト 172 を作成し、アサーションプロンプト 172 を計画生成機能 122 に送信する。プロンプト生成機能 120 は、環境 108 の現在の状態（例えば、フィードバック機能 126 から受信

した状態データ 160 を使用して決定される）に基づいてアサーションプロンプト 172 を作成する。

環境 108 の現在の状態には、エージェント 104 の現在の状態が含まれる。計画生成機能 122 は、アサーションプロンプト 172 に含まれるアサーションの真実性を判断または予測し、コンピューティングシステム 102 およびエージェント 104 によって実装されたタスク処理機能 124 に結果 174 を転送する。

次に、タスク処理機能 124 は、結果 174 に基づいて、エージェント 104 が実行する次のタスクを選択する。たとえば、計画生成機能 122 が、エージェント 104 がゴミ箱 164 に近いというアサーションが偽であると判断した場合、タスク処理機能 124 は、エージェント 104 をゴミ箱 164 に対して再配置する次のアクションを選択する。

一方、このアサーションが真である場合、タスク処理機能 124 は、リンゴ 162 をゴミ箱 164 に入れるなど、アサーションが真であることを条件とする次のタスクを選択する。したがって、アサーション 170 は、プロンプト生成機能 120、計画生成機能 122、および環境 108 の状態（たとえば、フィードバック機能 126 によって収集されたもの）を使用して、計画 152 内で条件ステートメントを実装できる。

図 2A はプロンプト 200 の例を示す。

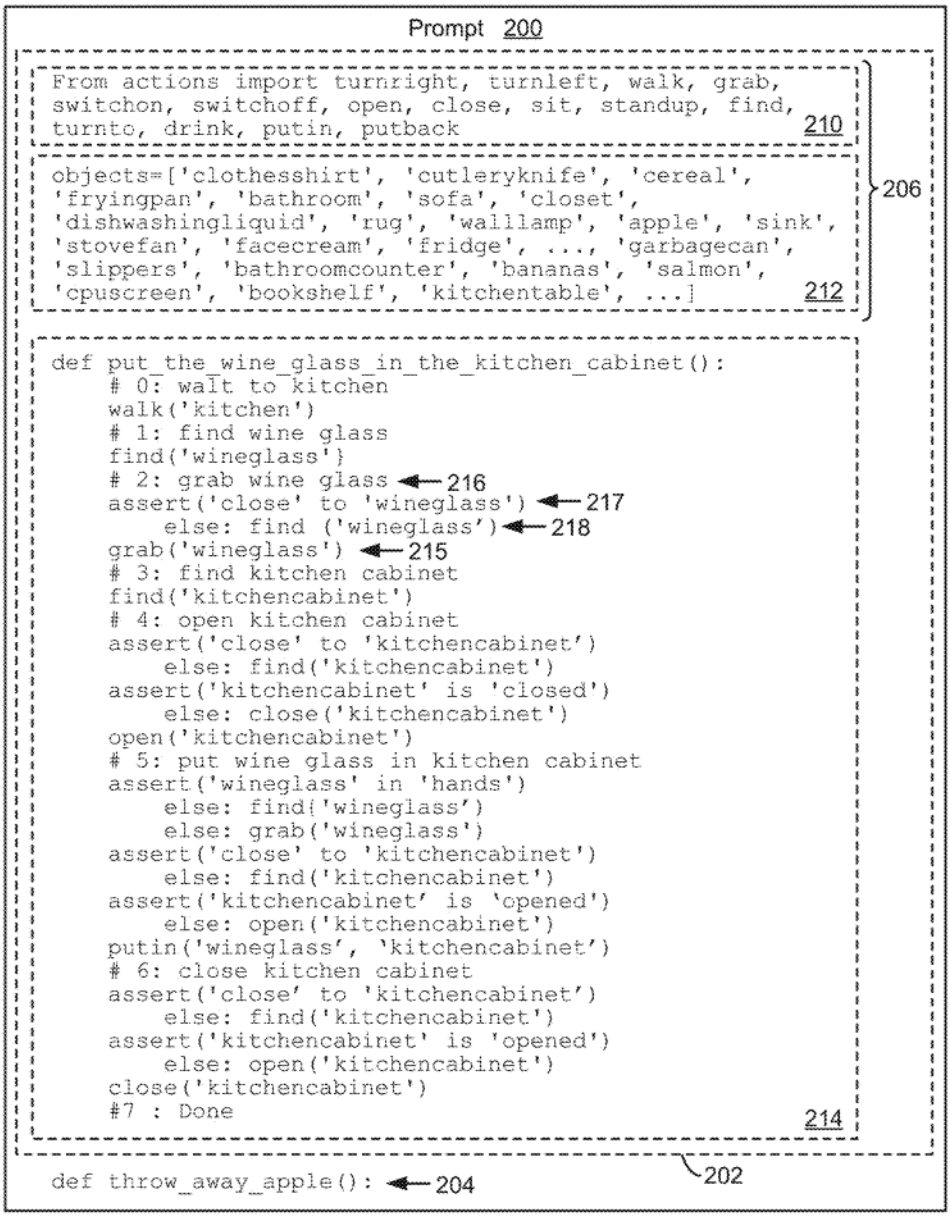


FIG. 2A

プロンプト 200 は、プロンプト生成機能 120（図 1 参照）によって作成される。プロンプト 200 は、コンピュータコード (例: Pythonic コード)として実装できる。プロンプト 200 には、操作情報 202 と、次の高レベルタスクの識別子 204 とが含まれる。

操作情報 202 には、環境情報 206 (たとえば、アクションプリミティブのセット 210 とオブジェクトリスト 212) と、タスク例 214 とが含まれる。アクションプリミティブのセット 210 は、エージェント 104 が実行できる 1 つ以上のアクションをインポートするためのコマンドとして実装できる。アクションプリミティブのセット 210 は、次

の形式を持つ文字列として実装できる：“from actions import<action primitive><#arguments>, <action primitive><#arguments>, ...”
アクションプリミティブのセット 210 は、計画生成機能 122 が、エージェント 104 が実行できないアクションを計画 152 に含めることを防ぐのに役立つ。

オブジェクトリスト 212 には、エージェント 104 の環境 108 内の任意のオブジェクトが含まれる。オブジェクトリスト 212 は、次の形式を持つ文字列のリストとして実装できる: “objects=[‘<object>’, ‘<object>’, ‘<object>’, . . .].”
オブジェクトリスト 212 は、計画生成機能 122 がエージェント 104 で使用できないオブジェクトを計画 152 に含めることを防ぐ。

タスク例 214 にはそれぞれ、コメントとして含まれる高レベルのサブゴールと、オブジェクトリスト 212 内の 1 つ以上のオブジェクトに関する関数呼び出し(function calls)としてオプションで含まれる実行可能環境アクションを含む関数定義が含まれる。タスク例 214 は、特定のアクションと特定のオブジェクトを使用して例のタスクを完了する方法を示し、計画生成機能 122 に、提供された環境情報 206 内に出力（たとえば、計画 152）を制限するように指示する。

プロンプト生成機能 120 には、タスク例 214 として使用できる人間のアノテーション（コメントなど）が付いた多数のタスク（たとえば、50 個の現実的な家庭のタスク）を含むデータセットが含まれるか、またはそのようなデータセットにアクセスできる。計画生成機能 122 は、タスク例 214 を使用して、自由形式言語のサブゴール（たとえば、コメントに含まれるもの）を、実行可能な環境アクションを含む 1 つ以上のタスクにマッピングする。

この例では、タスク例 214 には、関数を定義する関数定義
（“def put_the_wine_glass_in_the_kitchen_cabinet( )”）が含まれている。
関数定義には、関数の名前、関数が取るパラメータ、関数を実装するコード（ソース コードなど）が含まれる（たとえば、環境 108 内のオブジェクトに対して実行されるアクションなど)。

各タスク例 214 は、環境 108 内の利用可能なアクション（アクションプリミティブのセット 210 に含まれる）およびオブジェクト（オブジェクトリスト 212 にリストされている）を使用して、特定のタスクを完了する方法を示す。
タスク例 214 は、タスク名
(例: “def put_the_wine_glass_in_the_kitchen_cabinet( )”）と実行されるアクションと

の関係、および関連するアクションとオブジェクトに対する制限を示す。

プロンプト生成機能 120 は、タスク例 214 (例えば、クラウドソースの例から) を取得し、計画生成機能 122 によって以前に生成された計画 (例えば、それぞれの高レベルタスクを正常に達成したと判断された計画) を使用する。

タスク例 214 には、1 つ以上のタスク (例: “grab(‘wineglass’)”)、アクションを要約するコメント (例: “#2: grab wine glass”)、および実行を追跡するアサーション (例: 「“assert (‘close’ to ‘wineglass’)”) が含まれる。タスク (例えば、タスク 215) には、アクションプリミティブ (例えば、“grab(‘wineglass’)”」) への 1 つ以上のアプリケーションプログラミングインターフェイス (「API」) 呼び出しが含まれる。

コメント 216 は、後続のアクションシーケンスの自然言語による要約を提供する。コメントは、高レベルのタスクを論理的なサブタスクに分解するのに役立つ。この分割により、計画生成機能 122 は、タスクとサブタスクに関する知識を自然言語で表現し、計画を支援する。コメントは、計画生成機能 122 に当面の目標を通知するのに役立ち、一貫性のない、矛盾した、または繰り返しのある出力の可能性を減らす。

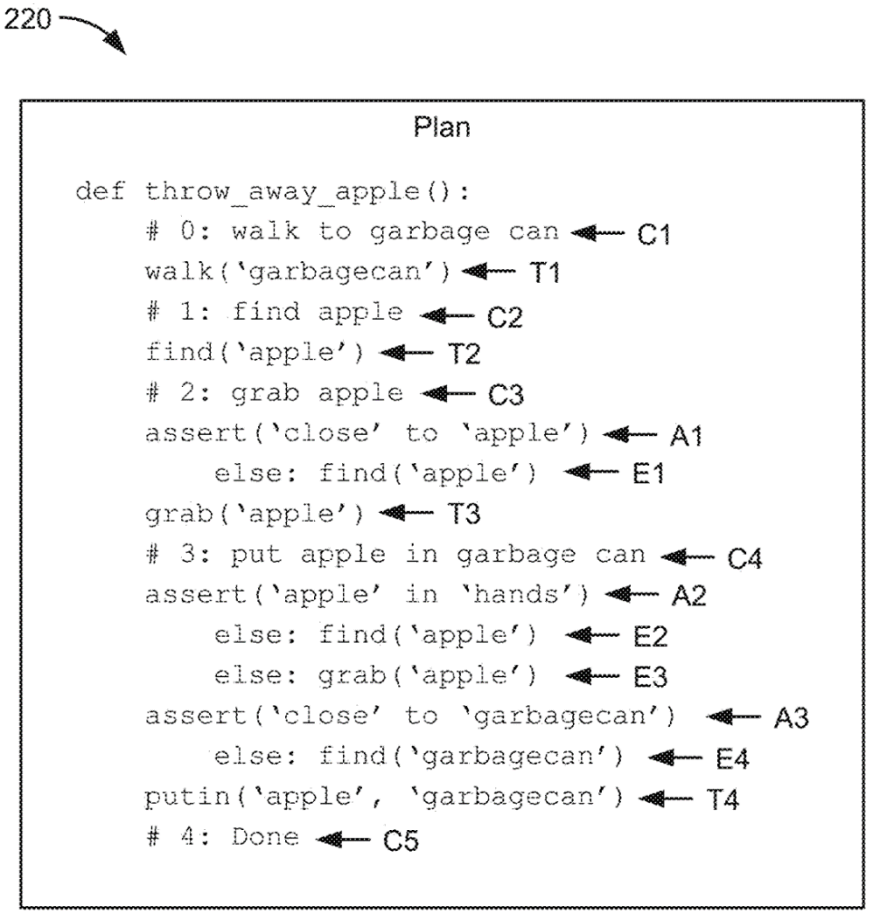
アサーション 217 は、次のアクションを実行するために必要な 1 つ以上の前提条件をアサートする。タスク例 214 のアサーションは、計画生成機能 122 への環境フィードバックメカニズムを示し、計画生成機能 122 が、前提条件が満たされることを保証する計画を作成するのに役立ち、前提条件が満たされていない場合にエージェント 104 が環境 108 を変更できるようにする (たとえば、エラー回復を有効にする)。

たとえば、アクション“grab(‘wineglass’)の前に、アサーション 217 はエージェント 104 がワイングラスに近いことをアサートする (例: assert (‘close’ to ‘wineglass’))。エージェント 104 がワイングラスの近くにない場合、エージェント 104 はまず、else ステートメント 218 (エラー回復ステートメントなど) でアクション“find (‘wineglass’)” を実行する。

アクション「(‘wineglass’)を見つける」は、エージェント 104 の少なくとも一部をワイングラスに近い位置に移動させ、アサーション 217 を True にする。タスク例 214 には、アサーション 170 に類似するが、1 つ以上の異なるアクションおよびオブジェクトを含むことができる 1 つ以上のアサーション (たとえば、アサーション 217) が含まれる場合があり、また、1 つ以上の回復モジュールが含まれる場合もある。

図 2A では、識別子 204 は、計画生成機能 122 に関数名「def throw_away_apple()」の関数定義を生成するように要求している。言い換えれば、プロンプト 200 により、計画生成機能 122 は、アクションプリミティブのセット 210、オブジェクトリスト 212、タスク例 214、および計画生成機能 122 が持つ可能性のある事前のトレーニングに基づいて、識別子 204 に続くコード (たとえば、ソースコード) (たとえば、識別子 204 内のコロンの後のコード(":") を予測する。

図 2B は、プロンプト 200 に応答して計画生成機能 122 によって生成された計画 220 を示している。



計画 220 は、コード (たとえば、Pythonic コード) として実装される。計画 220 には、計画生成機能 122 が図 2 Aのプロンプト 200 に少なくとも部分的に基づいて生成したタスク T1-T4 が含まれる。

タスク T1-T4 にはそれぞれ、アクションプリミティブのセット 210 内のアクションプリミティブと、オブジェクトリスト 212 からのオブジェクトが含まれている。たとえば、タスク T1-T4 には、それぞれ「歩く」、「探す」、「つかむ」、「置く」というアクショ

ンプリミティブが含まれており、それぞれが API 呼び出しになる。タスク T1-T4 には、環境 108 に存在するオブジェクト「ゴミ箱」と「リンゴ」も含まれている。

計画 220 には、コメント C1～C5 とアサーション A1～A3 が含まれる。アサーション A1 ～ A3 のそれぞれに、対応する else ステートメントが 1 つ以上ある。たとえば、アサーション A1 の後には else ステートメント E1 が続き、アサーション A2 の後には else ステートメント E2 と E3 が続き、アサーション A3 の後には else ステートメント E4 が続く。

計画生成機能 122 がアサーション A1～A3 のいずれかが False であると判定した場合、対応する else ステートメントが実行される。したがって、計画生成機能 122 が、アサーション A1 が False であると判定した場合、else ステートメント E1 が実行される。

同様に、計画生成機能 122 がアサーション A2 が False であると判断すると、else ステートメント E2 および E3 が実行され、計画生成機能 122 がアサーション A3 が False であると判断すると、else ステートメント E4 が実行される。else ステートメント E1-E4 を実行すると、アサーション A1-A3 の前提条件が満たされていることを確認できる。したがって、この例では、計画 220 によって、アサーション A1-A3 の前提条件が満たされた後にエージェント 104 がリンゴ 162 を投げ捨てる。

3.クレーム

077 特許のクレーム 1 は以下の通りである。

1. 方法において、

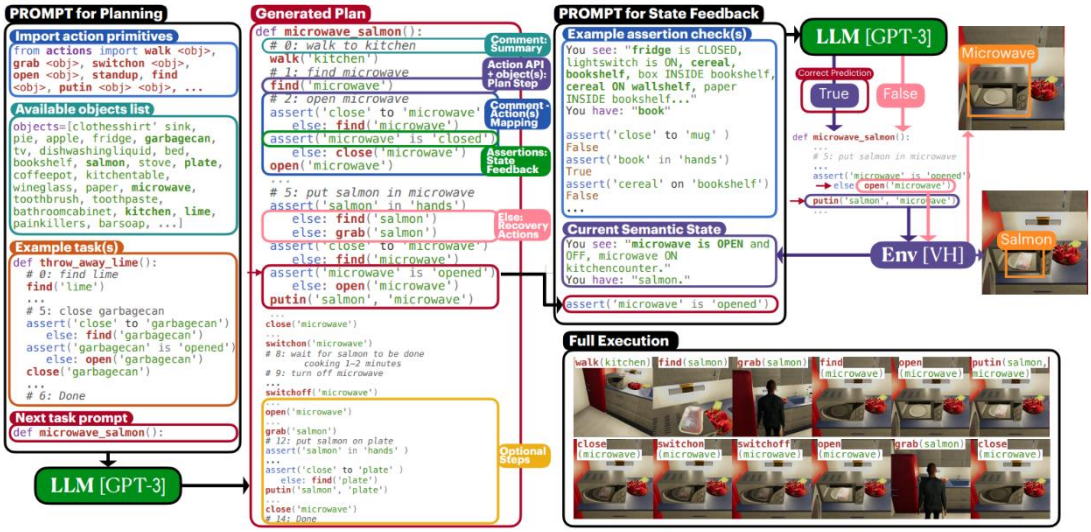
環境情報、少なくとも 1 つのタスク例、および環境内に存在するエージェントによって実行されるタスクの識別子を含むプロンプトを生成し、

機械学習プロセスにタスクを実行するための新しい計画を生成させるべく、少なくとも 1 つの機械学習プロセスにプロンプトを提供し、新しい計画は、環境内でエージェントによって実行可能な一連の操作で構成される。

4. 本特許に関連する論文

本特許に関する論文“PROGPROMPT: Generating Situated Robot Task Plans using Large Language Models” [1]が、NVIDIA の Valts Blukis 氏らにより公表されている。

[1] Valts Blukis, et al. “PROGPROMPT: Generating Situated Robot Task Plans using



論文の、プログラムプロンプトには、インポートステートメント、オブジェクトリスト、およびサンプルタスク（計画用の PROMPT）が含まれており、このプログラムプロンプトが LLM に与えられ、計画が生成される。生成された計画は、電子レンジでサーモンを調理することである。環境から得られる情報をもとに状態フィードバックプロンプトが生成され、このプロンプトも LLM に与えられる。

下記テーブルは、仮想的な家庭上で生成されたプログラムを評価したものである。

#	— Prompt Format and Parameters — Format	COMMENTS	FEEDBACK	LLM Backbone	*SR*	*Exec*	*GCR*
1	PROGPROMPT	✓	✓	CODEX	**0.40**±0.11	**0.90**±0.05	**0.72**±0.09
2	PROGPROMPT	✓	✓	DAVINCI	0.22±0.04	0.60±0.04	0.46±0.04
3	PROGPROMPT	✓	✓	GPT3	0.34±0.08	0.84±0.01	0.65±0.05
4	PROGPROMPT	✓	✗	GPT3	0.28±0.04	0.82±0.01	0.56±0.02
5	PROGPROMPT	✗	✓	GPT3	0.30±0.00	0.65±0.01	0.58±0.02
6	PROGPROMPT	✗	✗	GPT3	0.18±0.04	0.68±0.01	0.42±0.02
7	LANGPROMPT	-	-	GPT3	0.00±0.00	0.36±0.00	0.42±0.02
8	Baseline from HUANG ET AL. [2]			GPT3	0.00±0.00	0.45±0.03	0.21±0.03

PROGPROMPT は、利用可能な API に 2 つしか収まらない DAVINCI バックボーンを除いて、3 つの固定サンプルプログラムを使用する。[2]は、論文に記載されているように、動的に選択された 1 つのサンプルを使用する。LANGPROMPT は 3 つの自然言語テキスト例を使用する。

GPT3 バックボーンを使用した最高のパフォーマンスを発揮するモデルは青で表示され、全体的に最高のパフォーマンスを発揮するモデルは太字で表示される。PROGPROMPT の性能がベースライン[2]および LANGPROMPT を大幅に上回ってい

Large Language Models”　arXiv:2209.11302v1 [cs.RO] 22 Sep 2022

ることが理解できる。本研究に関する動画及びプログラムは下記サイトからアクセスすることができる。

progprompt.github.io

以上

著者紹介

河野英仁

河野特許事務所、所長弁理士。立命館大学情報システム学博士前期課程修了、米国フランクリンピアースローセンター知的財産権法修士修了、中国清華大学法学院知的財産夏季セミナー修了、MIT(マサチューセッツ工科大学)コンピュータ科学・AI 研究所 AI コース修了。

AI 特許コンサルティング、医療 AI 特許コンサルティングの他、米国・中国特許の権利化・侵害訴訟を専門としている。著書に「世界のソフトウエア特許(共著)」、「FinTech 特許入門」、「AI/IoT 特許入門 3」、「ブロックチェーン 3.0(共著)」がある。