

対話型生成AIにより大きく変わる エンジニアリングの世界

国立情報学研究所 石川 冬樹

f-ishikawa@nii.ac.jp / @fyufyu

<http://research.nii.ac.jp/~f-ishikawa/>

自己紹介

■ 国立情報学研究所 准教授

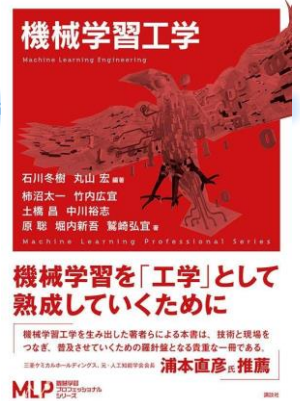
- ソフトウェア工学，特にディペンダビリティ：
形式手法，自動テスト生成，安全性論証など

■ 主な研究プロジェクト

- JST MIRAI-eAI：機械学習システムのディペンダビリティ
- JST ERATO-MMSD：自動運転システムの安全性

■ 産業界向け教育・実践研究

- トップエスイー，日科技連SQiP，電通大AISECなど
- 機械学習工学コミュニティ（MLSE研究会，QA4AI）



eAI



はじめに

LLMと対話型生成AI

■ LLM (Large Language Model, 大規模言語モデル)

- 言語モデルとは, 「入力された単語列に対しその続き (の確率)」を学習し, 生成することができるような機械学習モデル
- その大規模化が進んだ (後述)

■ 対話型生成AI

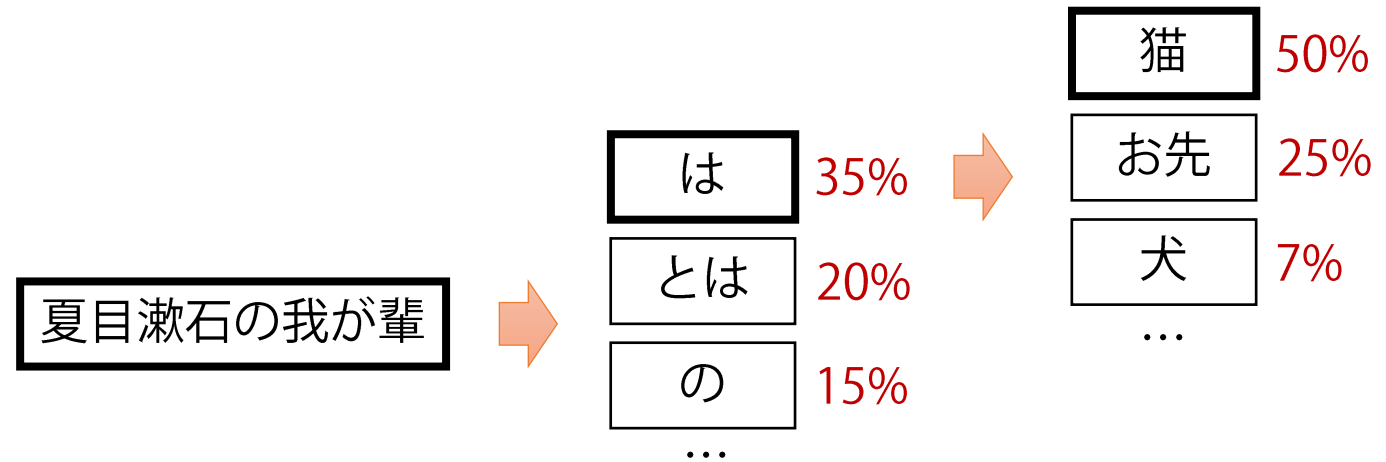
- LLMを活用することで, 自然な対話ができるようになったAIの総称
- ChatGPT (Open AI), Bard (Google) など

■ 補足

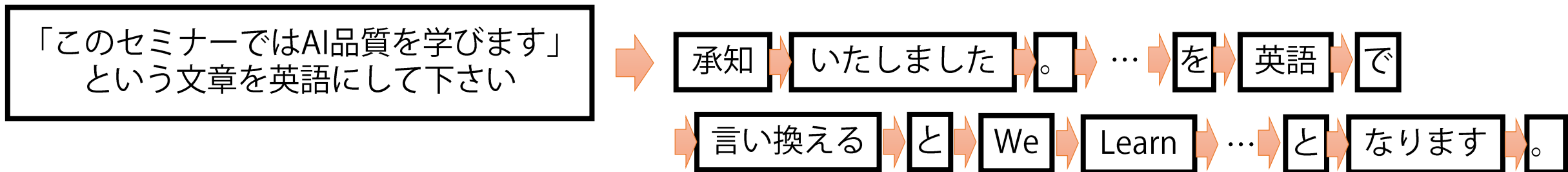
- 広く話題になっているChatGPTは, Open AIによる一つのサービスの名称
- 「生成AI」は, 「何がどれだけの確率で存在するか」などを学習することで, データを生成するAIの総称 (文章だけでなく, 画像や音楽などもある)

実現の雑なイメージ

- 言語モデルが学習し、出力できること



- これを大規模学習により極めていくと...



- 実際には、「返答が人間にとって望ましいかの判断基準」を別途学習するなどの工夫が加えられて実用的なものとなっている

すごいこと

■すごいこと

- 翻訳, 要約, 言い換え, 質問応答, プログラム作成・修正, プログラム説明など多様なタスクに利用でき, しかも従来の専用AIより良いことも多い
- 「…の立場で」のようなロールプレイング指示や, 返答に追加・修正を求める対話反復により, 意図した返答が得られるように調整できる
- 入出力が自然言語なので, 利用ハードルが低い

➡ 「システム開発」を経ずエンドユーザーが「自分用機能」をその場で得られる

■インパクト (無数に言われているがごく一部)

- ChatGPTのユーザー数は2か月で1億超え
- 産学・国内外多様なコミュニティで議論

一つの実行例



以下で述べる予防接種のシステムについて、考えられるユースケースをすべて時系列を明確にして述べてください。

利用者は市町村から発行された番号でログインしてワクチン接種の予約を行う。システム上でのキャンセルは2日前まで可能でそれ以降は病院への電話となる。ワクチンにはいくつか種類があり、病院によって対応しているワクチンが異なる。利用者は、予約を希望する病院、ワクチンまたは日時から検索して、ワクチン接種の予約を行うことができる。



以下は述べられた予防接種のシステムのユースケースを時系列に沿って説明したものです。

1. 利用者登録と認証の時点:

- 利用者が市町村から発行された番号を受け取る。
- 利用者がシステムにログインし、発行された番号を使って認証を行う。

2. 予約検索の時点:

- 利用者が接種を希望するワクチンの種類、病院、または日時から検索開始。
- システムが利用者に対応しているワクチンのある病院のリストを提示。

3. 予約の時点:

- 利用者が特定の病院、ワクチン、日時を選び、予約を行う。
- システムが予約確認の通知を利用者に送る。

4. キャンセルの時点（2日前まで）:

- 利用者が予約をキャンセルしたい場合、システム上で2日前までキャンセル可能。
- システムがキャンセル確認の通知を利用者に送る。

5. キャンセルの時点（2日以降）:

- 2日以内でキャンセルしたい場合、利用者が病院へ直接電話。
- 病院がシステム上でキャンセルを反映。

6. 接種の時点:

- 利用者が予約した日時に病院で接種。
- 病院がシステム上で接種情報を更新。

7. 接種記録の時点:

- 利用者がシステムにログインして自身の接種記録を確認。
- 必要に応じて利用者が接種証明書などのドキュメントをダウンロード。

このシステムは、予約から接種に至る一連の流れをデジタル化し、利用者や病院とのコミュニケーションを促進します。特定のワクチンの供給状況や、地域の病院における利用可能な予約枠などを効率的に管理するために使用されると考えられます。

2023/11/22

修正指示もできる



接種証明書はシステムで扱わないように変更してください。



もちろんです。接種証明書をシステムで扱わないユースケースを以下に記載します。

1. 利用者登録と認証の時点:

- 利用者が市町村から発行された番号を受け取る。
- 利用者がシステムにログインし、発行された番号を使って認証を行う。

2. 予約検索の時点:

- 利用者が接種を希望するワクチンの種類、病院、または日時から検索開始。
- システムが利用者に対応しているワクチンのある病院のリストを提示。

「壁打ち」・アイデア出しの例



このシステムについて、利用者にとって不便な点を挙げて下さい。



このワクチン接種の予約システムにおいて、利用者にとって不便な点は以下のようなものが考えられます。

1. ログイン認証の手間:

- 市町村から発行された番号を使う認証方式は、番号を紛失したり忘れたりした場合にログインできなくなる恐れがある。

2. ワクチンや病院の選択の複雑さ:

- 複数のワクチン種類と病院の組み合わせがあるため、どのワクチンがどの病院で接種できるのかを調べるのが複雑に感じる場合がある。

3. 2日以内のキャンセル手続き:

- 2日以内であれば病院への直接電話が必要なため、手間がかかることがあり、言語のバリアや通話が苦手な人にとっては障壁になる可能性がある。

4. システムの使い勝手:

- インターフェイスが使いにくい、検索機能が不足している、などのシステムの操作性に関連する問題も不便に感じる可能性がある。

5. 予約の可用性と情報の透明性:

- 人気の病院や希望するワクチンがすぐに埋まる可能性があるため、予約の可用性に不満を感じることもあるかもしれない。
- また、各病院のワクチン在庫や予約状況などの情報が不足していると、適切な選択を行いにくい場合がある。

6. 技術的な障壁:

- インターネットへのアクセスが制限される地域や、コンピュータ操作が得意な高齢者などにとっては、オンラインでの予約システム自体が障壁になる可能性がある。

7. プライバシーとセキュリティの懸念:

- 市町村からの番号発行や、個人情報やオンラインで取り扱うプロセスにおいて、セキュリティやプライバシーの懸念が利用者にとって不便や不安をもたらすことがある。

これらの不便な点を解消するためには、ユーザビリティの向上、システムの透明性とセキュリティの強化、多様な利用者ニーズに対応したサポート体制の構築などが求められるでしょう。

限界

- ルールや知識に基づいて処理しているわけではない
 - 数学や論理, 事実関係や知識の問題については, どこかに限界がある
 - 「もっともらしい嘘」をつくことがある (ハルシネーション問題)
 - 「有名人Aの親の名前はB」と答えられても, 「Bの子供は?」となると答えられない (知識照らし合わせではなく典型で答えているため)
- 訓練データのカバー範囲や偏り (バイアス) に影響を受ける
 - 「消防士だと男となる割合が高い」といった偏りがあるかもしれない
 - アメリカで作られたLLMは, 日本語より英語で入力した方が返答の質が良い
- LLMを一から構築できる組織は限定的 (コスト・スキル)
 - 数十万円から百万円以上するGPUを多数 (ときには百~千個), 何日間も
 - 訓練時間の報告はよく見かけるが, 実際は試行錯誤が必要なはず

限界？

- ・・・のようなことはよく言われるが
 - 大規模学習において得られた「ありがちな回答」は、「論理的な回答」であり、あたかも論理的思考があるように見える回答が出てくることが多い
 - この程度がものすごいので、何か「境界線」を超えたような気もする
 - 検索エンジンや計算ツール、知識データベースなど外部ツールと組み合わせることで、最新の知識や正確さを反映するように補完できる
 - 例：RAG (Retrieval-augmented Generation)
 - ChatGPTは控えめ（何かを断言しない）になる調整がされた模様
 - 差別や人格攻撃などの不適切な発言は避ける調整は注意深くされているはず
 - 一から作ることは難しくても、固有のデータを加えての追加学習や、他機能との組み合わせなら各組織で十分できる

「嘘をつく」例とその変遷

それらしい嘘をつく例



東京特許許可局について教えてください。



東京特許許可局は、日本の特許庁の管轄下にある特許審査・登録の申請受付、審査業務を行う機関の一つです。東京都千代田区にあります。

東京特許許可局では、特許出願や商標出願などの申請を受け付けており、審査官による書類審査や実用新案・意匠の登録審査などを行っています。また、特許出願においては、出願の受付から公開までの手続きも行っています。

東京特許許可局は、日本の特許制度の中心的な機関の一つであり、企業や個人の知的財産権保護に重要な役割を果たしています。



東京特許許可局の最寄り駅はどこですか？



東京特許許可局の最寄り駅は、JR総武線「飯田橋駅」、東京メトロ東西線・有楽町線・南北線「飯田橋駅」、東京メトロ半蔵門線・南北線「永田町駅」、都営地下鉄三田線「半蔵門駅」のいずれかになります。飯田橋駅から徒歩で約10分程度、永田町駅・半蔵門駅からは徒歩で約15分程度の場所にあります。



Regenerate response

駅と路線の対応や距離関係がおかしい

[ChatGPT Feb 13 ver. (GPT-3.5)]

Default (GPT-3.5)



東京特許許可局について教えてください

最近はお断りの言葉が入る



申し訳ありませんが、私の知識は2021年9月までのものであり、それ以降の情報は持っていません。したがって、2023年の情報については提供できません。ただし、一般的に東京特許許可局（Tokyo Patent Office）についてお話しします。

東京特許許可局は、日本における特許や実用新案、意匠などの知的財産権に関する申請や登録、審査を担当する機関です。特許許可局は、日本国内の各地に設置されていますが、東京特許許可局はその中でも最も重要な局の一つとして知られています。東京特許許可局の管轄区域は東京都を中心とした広範囲に及びます。

特許許可局は、特許庁（Japan Patent Office, JPO）の下部組織として運営されており、知的財産制度の維持と発展に向けた重要な役割を果たしています。特許許可局は、特許法や実用新案法などに基づいて、新規性や非自明性、産業上の適用可能性などを審査し、特許権の付与を決定します。

具体的な最新情報については、日本特許庁の公式ウェブサイトや関連する公式情報源を確認してください。

[ChatGPT Jul 20 ver. (GPT-3.5)]

GPT-4



東京特許許可局について教えてください

最新版では学べている



「東京特許許可局」は、実際には存在しない架空の組織で、日本語の早口言葉として有名です。このフレーズは、「とうきょうとつきよきよかきよく」と発音することから、日本語を母国語とする人々でも発音が難しいとされることが多いです。

早口言葉は、言葉の発音を速く連続して行う練習やゲームとして用いられることが一般的で、発音の練習や、言語のリズムと流れを楽しむために使われます。「東京特許許可局」のようなフレーズは、子供から大人まで楽しむことができる、日本語の言語文化の一部です。

[ChatGPT Jul 20 ver. (GPT-4)]

本講演の焦点

- ソフトウェア工学に関する二つの観点から整理, 議論
 - LLM・対話型生成AIのためのソフトウェア工学的な方法論・技術 (SE4LLM/AI)
 - ソフトウェア工学におけるLLM・対話型生成AIの活用 (AI4LLM/SE)
- ※ 私からはクラウド・セキュリティに限らない広い話

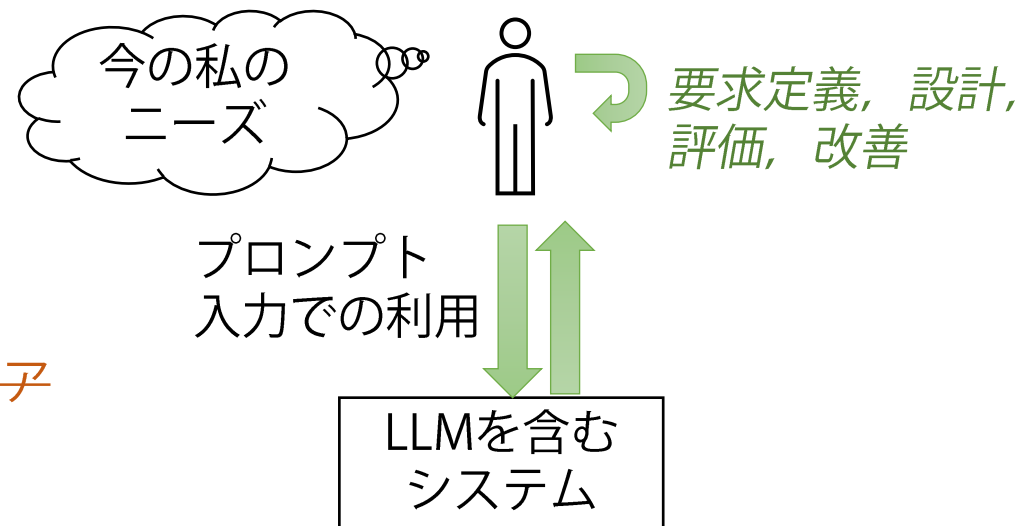
- その他幅広い議論を引き起こしているものの, 今日は対象外
 - 我々が「知的な仕事・活動をしている」と思っていたものは, ありがちな「典型」を繰り返していただけだった…?
 - 変なことを聞いても馬鹿にされず, 何を言っても真摯に答えてくれるのは, 対話・質問の相手として優れている・心理的に嬉しい
 - ○○○の職業がいらなくなるのでは?
 - …

SE for LLM/AI

そもそも何をエンジニアリングするのか？ (1)

- 極端な場合：各自がいろいろなタスクに自由にLLMを使う
 - ドメイン・タスク（ユースケース）を固定せず，利用者がプロンプトを入力
 - 「要求の範囲」や「開発時」が存在しない
- ➡ ソフトウェア工学の原則・知識・技術は役に立つ…？

各自利用型



➡ 事前・開発時 by エンジニア

➡ 利用時 by 利用者

【補足】よく知られる取り組み：Prompt Engineering

■ Prompt Engineering Guideより引用

[<https://www.promptingguide.ai/jp>] (2023/6/3アクセス時)

■ 例：Few-shot prompting + Chain-of-thought prompting

プロンプト:

このグループの奇数を合計すると偶数になります。： 4、 8、 9、 15、 12、 2、 1。
A: 奇数を全て加えると (9, 15, 1)25になります。答えはFalseです。
このグループの奇数を合計すると偶数になります。： 17、 10、 19、 4、 8、 12、 24。
A: 奇数を全て加えると (17, 19)36になります。答えはTrueです。
このグループの奇数を合計すると偶数になります。： 16、 11、 14、 4、 8、 13、 24。
A: 奇数を全て加えると (11, 13)24になります。答えはTrueです。
このグループの奇数を合計すると偶数になります。： 17、 9、 10、 12、 13、 4、 2。
A: 奇数を全て加^{LLMのトレンド}ると (9, 13)39になります。答えはFalseです。
このグループの奇数を合計すると偶数になります。： 15、 32、 5、 13、 82、 7、 1。
A:

具体例を見せる

途中過程を示す

出力:

奇数を全て加えると (15, 5, 13, 7, 1)41になります。答えはFalseです。

そもそも何をエンジニアリングするのか？ (2)

- もう片方の極端な場合：LLMを含めたシステムを開発する
 - ドメイン・タスク（ユースケース）を固定
 - 利用者は定型入力を行い，システム内部でプロンプトを作りLLMを利用
 - ➡ 「出力が不定形」などLLM固有の特性があるが，従来のシステム開発に近い

従来プロジェクト型



例：予約サイトの口コミ要約にだけ
LLMを内部的に利用

抽出
定義

利用

要求仕様

LLMを含む
システム

ドメイン固有
プロンプト
生成パターン

➡ 事前・開発時 by エンジニア

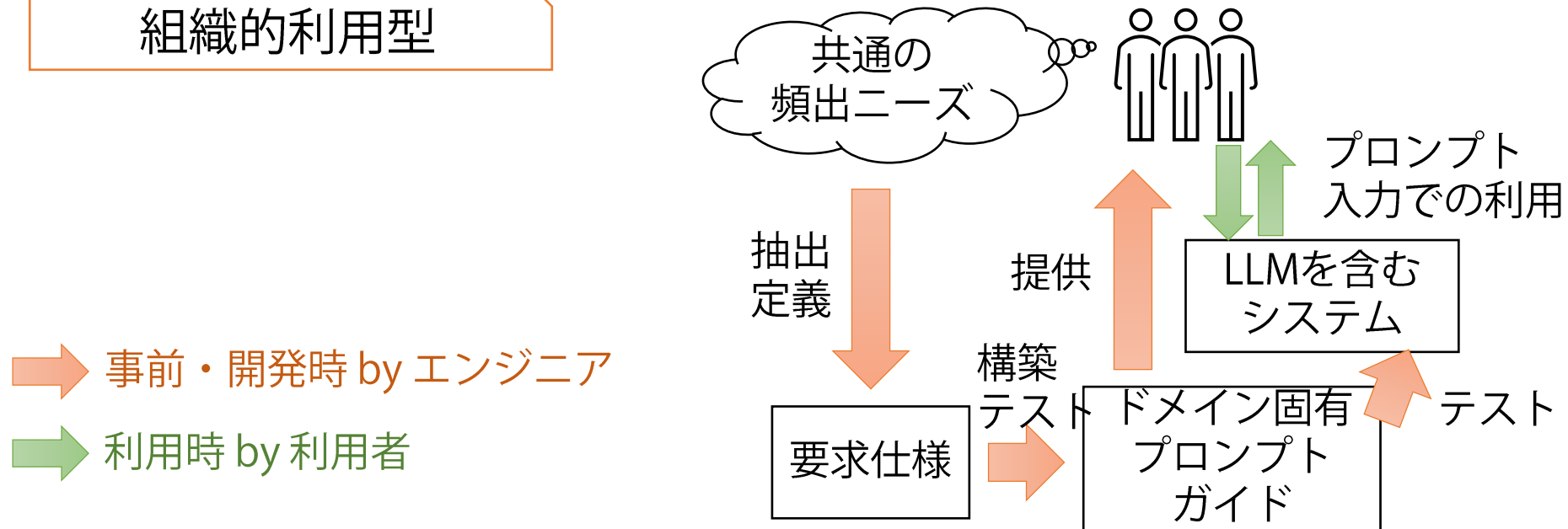
➡ 利用時 by 利用者

構築
テスト

そもそも何をエンジニアリングするのか？ (3)

- 中間：事前にタスクを決め，LLMの利用法を設計・評価して使う？
 - ドメイン・タスク（ユースケース）を固定
 - 事前に，固有のプロンプトガイドライン構築やLLMの十分性評価を行う
 - 利用者は，一定範囲のプロンプトを用いてLLMを利用
- ➡ プロセスは従来に近いが「システム開発」ではなく「ソリューション整備」

組織的利用型



【補足】 GPTs

- Open AIにおいて11月発表の新機能：
ChatGPTのカスタム版のようなものを作りやすく
 - ファイルをアップロードして読ませたり
 - 対話によりどういう振る舞いをさせたいか指示したり



You

クラウドの規約についてISO/IEC 27017に準拠しているか検査するGPTを作りたい。与えられた規約について、ISO/IEC 27017の各項目に準拠しているかを確認して、その結果を報告する。準拠したあるいはしなかったと判断した理由を添えること。



GPT Builder

Great! The GPT will inspect cloud service terms and conditions for compliance with ISO/IEC 27017, analyzing provided terms to determine if they meet the standards set by each clause of ISO/IEC 27017. It will report the results with detailed explanations for compliance or non-compliance, while maintaining a professional and objective tone.

Now, let's give this GPT a name. How about "Cloud Compliance Inspector"? Do you like this name, or would you prefer something different?



You

Yes, that name is good

この後シンボルとなる図が作られて、いくつかの質問に答えていき設定完了

- ユースケース
- 重視すること、回答に含めるべきでないこと
- あいまいな記述を適当に補うか明確化依頼するか
- どれくらい形式的な言い回しで答えるべきか

2023/11/13時の試行

ソフトウェア工学の役割？

- 現在まず広く飛び交っている話題は「各自利用型」
 - 「従来プロジェクト型」「組織的利用型」についても進行中の模様
 - ある種のデザインパターンと評価についてはすでに活発な議論
 - 例：Azure OpenAI Design Patterns [<https://github.com/microsoft/azure-openai-design-patterns>]
 - 「長文の要約には、部分ごとの要約を作らせ、それらの要約を作らせる」など
 - 例：多様なタスクでの評価 [Bang et al., A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity, arXiv'23]
- ➡ 技術的にはソフトウェア工学も貢献できる！
- システム開発が伴う場合はもちろん！
 - 直接LLMを使う場合も「ソリューション開発」への原則・技術が生きるはず

LLM/対話型生成AIにおける品質特性？

- 行われてきた評価やISO 25029:2023 (SQaRE for AI) を踏まえると…
 - 正確さ：想定機能・タスクに対して
 - 事実性：検証可能な事実や自身の返答における整合
 - 頑健性：未知や敵対的，不正な入力への耐性
 - 倫理性：公平性と安全性など
 - 制御可能性・協調可能性：指示に応じて返答が適切に変わる
 - セキュリティ：敵対的な入力，訓練パイプラインやモデルへの攻撃対処など
 - 「トラスト」に関する観点はもっと広い？
 - 返答の多様性，返答における個性，…？

➡ QA4AIガイドラインの一部として近日中に発行予定

LLM/対話型生成AIにおける品質評価手法？

- これまでの（教師あり学習型）AIにおける課題をさらに難しく
 - テスト・評価の成否判断やスコア付けがあいまい・高コスト（いわゆる Non-Testable ・オラクル問題）
 - バージョン比較や回帰テストにおけるテスト成否の基準は？
 - 最初の返答の正否だけでなく，対話により修正する能力などをどう測る？
 - 入出力ともに不定形であり，要求やその境界があいまいで，「ユースケースに一通り対応」の定義や，評価の定式化・自動化が難しい
 - …

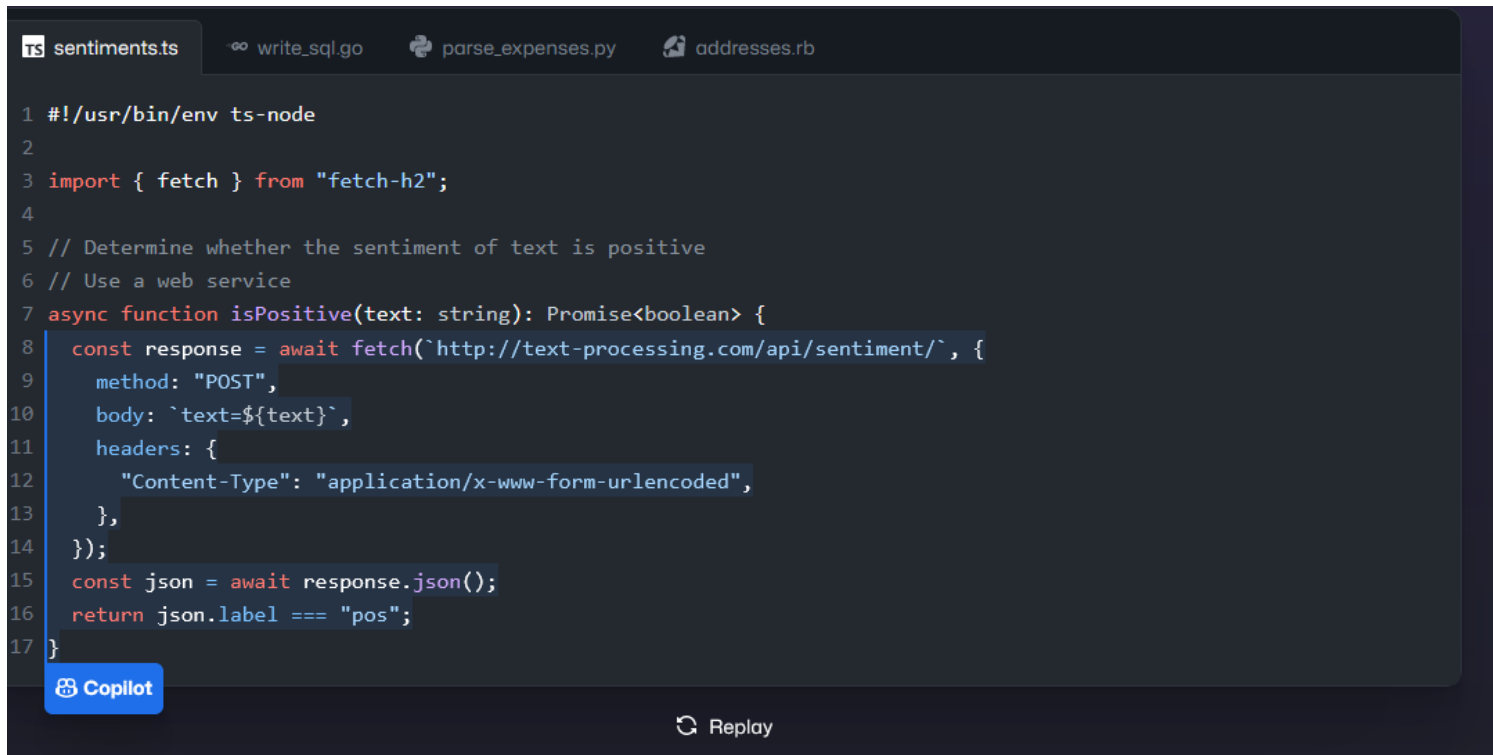
➡ 様々な評価がすでになされているが（論文媒体で200件以上），現時点では標準的な評価プロトコルは確立されていない

[Chang et al., A Survey on Evaluation of Large Language Models, arXiv'23]

LLM/AI for SE

■ Generative Chat AI for SE

- すでに多数の試み
- コーディング補助はメジャー（GitHub Copilot等）



```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

Copilot

Replay

[<https://github.com/features/copilot/>] (2023/6/3アクセス時)

- ブログ等でも多数の試みやガイドが観測されている
 - コーディング関連だけでなくすでに多数の試み
 - 要求分析：「大まかな案件」から「機能要件」や「ユースケース図」を作成
 - [<https://zenn.dev/yamadamadamada/articles/40d0594c1c6375>]
 - 設計：設計パターンの比較・「壁打ち」
 - [https://dev.classmethod.jp/articles/software_architecture_conversation_with_ai/]
 - テスト：ケース作成・レビュー
 - [<https://prtn-life.com/blog/chatgpt-unittest>]
 - あらゆるタスクで活用可能性あり？

論文の例：単体テスト生成

■ ChatGPT vs SBST: A Comparative Assessment of Unit Test Suite Generation (arXiv, 2023)

- 正しく動くテストケースを作れる？ → まあまあ（70%+IDEで直せる程度）
- 理解しやすい？ → ととても、複雑さが低い、インデントやJavaDocは悪い
- コードカバレッジは？ → SBSTの方が高いが、ChatGPTも次点で悪くはない
- バグ検出能力は？ → SBSTの方が高い、
ChatGPTが作るアサーションは信頼できない

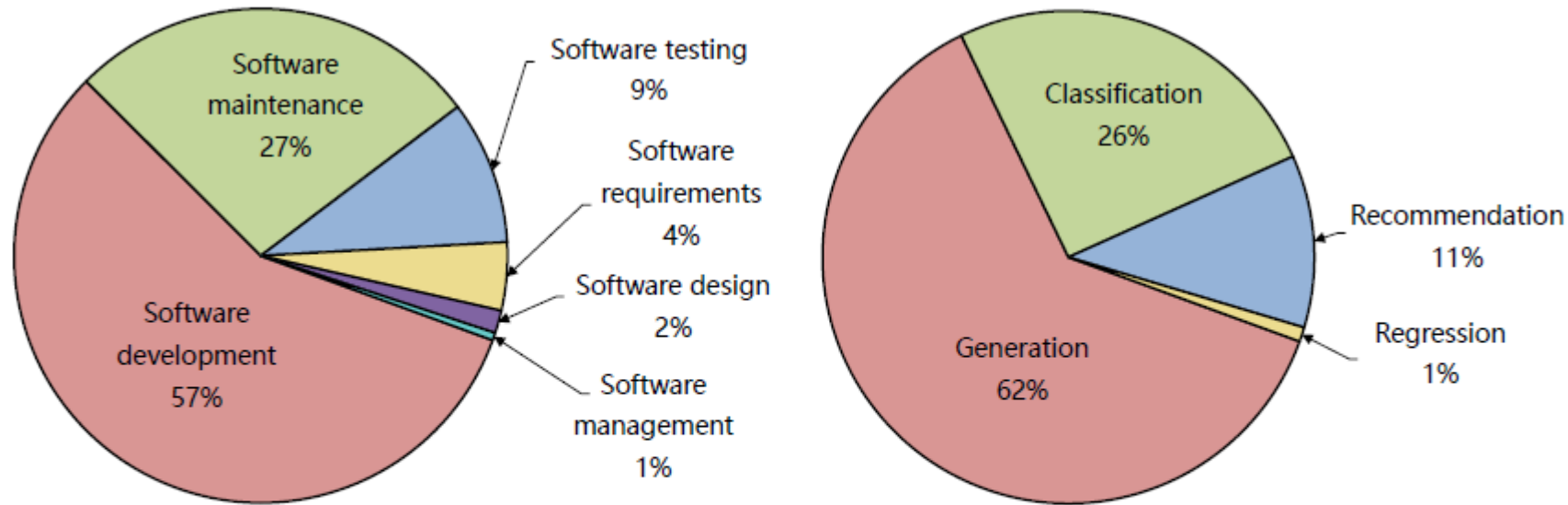
（ChatGPTもSBSTのような探索・反復プロセスで使えばよいのかも？
組み合わせる価値はある？）

論文の例：コード生成におけるChatGPTの非決定性

- LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation (arXiv, 2023)
 - Semantic difference（実行結果が変わるようなコード差異）が複数解の生成で観測される
 - 文法的にも
 - 俗に言われるTemperatureの設定であまり非決定性は排除できない
 - Top-1を5回出すのと、Top-5を1回出すのとで同じようにばらつく
 - 指示が長いほどバラツキが多く、かつバグがあることが多い

論文の例：Systematic Literature Survey

■ 2023年時点での取り組みの様子



(a) Distribution of LLM usages in SE activities. (b) Problem classification based on collected studies.

Fig. 7. Distribution of LLM utilization across different SE activities and problem types.

[Hou et al., Large Language Models for Software Engineering: A Systematic Literature Review, arXiv'23]

おわりに

全体まとめ：「AIソフトウェア」の品質

- これまでの教師あり学習型AIでの意識がさらに強く・広く：
高い**不完全さ・不確実性**

- 要求・環境・傾向の膨大さ + 実装した挙動の理解困難性
- 実世界や社会に大きく踏み込むことが増加
- 不確かなものに対する試行錯誤の反復継続
- めまぐるしく変わる技術と世界

ステークホルダーと議論し不完全・不確かなものを受け入れ
自動化された探索・テスト・測定・監視による継続的進化を
&
新しい技術・時代の変化を楽しみましょう！