

テキスト生成 AI 利活用におけるリスクへの対策ガイ
ドブック（ α 版）

2024（令和6）年 5月 29日

デジタル庁

〔ドキュメントの位置付け〕

参考資料。今後、デジタル社会推進標準ガイドラインへの編入を検討予定

〔キーワード〕

テキスト生成 AI、生成 AI、サービス開発者、サービス提供者

〔概要〕

テキスト生成 AI を利活用し、行政サービスや職員業務の改善の重要度が高まる中、リスクを特定し、そのリスクを受容できるレベルまでに軽減する対応もまた重要になっている。テキスト生成 AI に関連するリスクは多岐にわたるが、その多くはテキスト生成 AI 固有でない AI システム全般に共通するものである。そこで、本文書では政府情報システムを対象に、テキスト生成 AI 固有と見られるリスクに焦点をあて、留意点を紹介する。現段階（2024 年 5 月現在）では、実践的なフレームワークやチェックリストによるガイドブックを作成できるまでの知見が蓄積されていないため、留意点の紹介にとどめる。本文書は、テキスト生成 AI をその提供形態と利活用方法で分類し、それぞれにおける想定リスクとその軽減策を具体的に示したものである

改定履歴

改定年月日	改定箇所	改定内容
2024年5月29日	-	初版決定

目次

目次	1
1 はじめに	4
1.1 目的とスコープ	4
1.2 本文書で想定するテキスト生成 AI の利活用方法	6
1.3 本書の構成	6
1.4 用語	6
2 新サービス企画時のテキスト生成 AI 固有の留意	8
2.1 テキスト生成 AI の利活用が不適切であるケースにテキスト生成 AI を用いるリスク	8
2.2 テキスト生成 AI の利活用が効果的なケースを見落とすリスク	9
1) 情報検索サービスを国民向けに展開する事例でテキスト生成 AI を適切に利活用するユースケース	9
2) パブリックコメントの返答作成業務にテキスト生成 AI を適切に利活用するユースケース	11
3) 法令で決められた資格保有者等の支援についてテキスト生成 AI を適切に利活用するユースケース	11
4) 信頼できるデータベースと連携してテキスト生成 AI を適切に利活用するユースケース	11
2.3 技術的実現可能性の検討によるリスク軽減	12
2.4 民間サービスの活用	16
2.5 2章のまとめ	18
3 予算要求前時のテキスト生成 AI 固有の留意点	19
3.1 コスト削減に向けた検討	19
1) ハードウェア・ソフトウェアのコスト削減観点	19
2) アプリケーションのコスト削減観点	20
3) 運用業務のコスト削減観点	21
4) その他のコスト削減観点	22
3.2 3章 まとめ	23
4 調達実施前時のテキスト生成 AI 固有の留意点	24
4.1 チャットでの利用の期待品質	24
4.2 バッチ処理での利用の期待品質	25
4.3 情報検索での利用の期待品質	26
4.4 自然言語からのプログラム作成での利用の期待品質	26

4.5	4章 まとめ	26
5	設計・開発時のテキスト生成 AI 固有の留意点	28
5.1	テキスト生成 AI の生成物の品質面への懸念	28
	1) テキスト生成 AI を分類やラベリングに利活用するケース	28
	2) テキスト生成 AI の生成物を文章として扱うケース	28
5.2	テキスト生成 AI による生成物の品質評価・品質保証についてのリスクとその軽減策	29
	1) 生成結果がばらつくことに起因するリスク	30
	2) 生成物の品質評価が困難であることに起因するリスク	31
	3) テストケースのカバレッジが不十分なことに起因するリスク	33
	4) テスト済みの生成物のみを用いる場合の工夫	35
5.3	5章 まとめ	36
6	サービス実施時に留意すべきテキスト生成 AI 固有の留意点	37
6.1	業務の実績データや利用者要望を分析できているか	37
	1) チャットでの利用の場合	37
	2) バッチ処理での利用の場合	37
	3) 情報検索での利用の場合	37
	4) 自然言語からのプログラム作成利用の場合	38
6.2	非機能要件に関する実績データを把握しているか	39
	1) チャット利用の場合	39
	2) バッチ処理での利用の場合	39
	3) 情報検索での利用の場合	39
	4) 自然言語からのプログラム作成での利用の場合	39
6.3	業務で扱うデータの品質を維持しつつ適切なライフサイクル管理を行っているか	40
6.4	6章 まとめ	40
	まとめ	41
	付録	42
7	提供形態ごとの想定リスク	42
7.1	テキスト生成 AI を組み込んだサービスとして提供されるケースに関するリスクとその軽減策	42
	1) ライセンスに関するリスク	42
	2) 調達容易性に関するリスク	42
	3) 機密性に関するリスク	43
	4) アップデート対応に関するリスク	43
	5) ベンダーロックインに関するリスク	43

6) コストマネジメントに関するリスク	44
7.2 テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケースに関するリスクとその軽減策	44
1) ライセンスに関するリスク	45
2) 調達容易性に関するリスク	45
3) 機密性に関するリスク	45
4) アップデート対応に関するリスク	45
5) ベンダーロックインに関するリスク	46
7.3 テキスト生成 AI の機械学習モデルが直接提供されるケースに関するリスクとその軽減策	48
1) ライセンスに関するリスク	48
2) 調達容易性に関するリスク	49
3) 機密性に関するリスク	49
4) アップデート対応に関するリスク	49
5) ベンダーロックインに関するリスク	49
6) コストマネジメントに関するリスク	49
8 従来型の情報検索サービスをテキスト生成 AI により改善する手法 ...	50
9 文章間の類似性の機械的評価方法について	55

1 はじめに

テキスト生成AIの利活用について社会的期待が高まる一方で、その様々なリスク（運用リスク、情報セキュリティ、プライバシー侵害やリーガルリスク等）への対応もその重要性が高まっている。その中でもテキスト生成AI利活用時の情報セキュリティに関する検証項目の多くは既存の情報システムやAIシステムと共通するが、一方でテキスト生成AI固有と思われるリスクも存在する。このテキスト生成AI固有のリスクは、そのテキスト生成AIのサービスの提供形態やユースケースに特異な場合もあり一概に議論することは困難である。

本文書は、[AI事業者ガイドライン（第1.0版）](#)（以降、AI事業者ガイドラインと称する）およびその[別添](#)（以降、別添と称する）を前提に、[DS-120 デジタル・ガバメント推進標準ガイドライン実践ガイドブック](#)（以降、実践ガイドブックと称する）への対応と実務上必要と思われる具体的なリスク軽減策について紹介する。重複した記述を出来る限り避けるために引用を多めにした構成にしたため、手元にAI事業者ガイドライン、別添と実践ガイドブックを用意してから読み進めてほしい。この構成は、テキスト生成AIを政府情報システムへの利活用を計画しているシステム管理者にとって参照しやすい実践的な内容になると考えている。

これらのリスクについての議論やその軽減策の方法論については、世間的にも発展途上のものである点を留意いただきたい。

1.1 目的とスコープ

本文書は、テキスト生成AIを活用した政府情報システムの開発及び業務改善に従事する関係者に対して、想定されるリスクの内、テキスト生成AI固有と考えられるリスクを特定し、そのリスクに対する軽減策の一般論を示すことを主な目的としている。

テキスト生成AIを活用したプロジェクトのステークホルダー全体は、[機械学習品質マネジメントガイドライン 第4版 Annex](#) (2,3ページ)によると以下の表のように分類される。

ステークホルダーの種別	役割
基盤モデル開発者	基盤モデルの訓練を行う。

サービス開発者	訓練済み基盤モデルを調整し、また、目的のタスクを基盤モデルが実行するよう誘導するプロンプトを装備することにより、目的とする用途向けのサービスを開発する。
サービス提供者	目的とする用途向けのサービスを運用する。
サービス利用者	上記サービスの直接利用者である。サービス利用者には個人と組織がある。組織の場合には組織内で使う場合と顧客向けに使う場合がある。
サービス利用者である組織の顧客	直接または間接にサービスを利用することがある(サービスを組織が個別の顧客向けに使う場合など)。
サービスを利用しない一般市民	サービス利用者やその顧客の行動によって影響を受けることがある。
行政等のガバナンス機関	サービスの社会的な影響を制御する(予防、軽減を図る、監視する、被害を補償する等)。

表1 機械学習品質マネジメントガイドライン 第4版 Annex での生成AIのステークホルダーの識別表

本文章はこの分類表の定義に従う。本文章の想定読者の役割は、(1) 訓練済み基盤モデルを調整し、また、目的のタスクを基盤モデルが実行するよう誘導するプロンプトを装備することにより、目的とする用途向けのサービスを開発する、と(2) 目的とする用途向けのサービスを運用する、を実施もしくは委託するサービス開発者とサービス提供者の2つである。本文章では基盤モデル開発者や行政等のガバナンス機関等の役割は想定していない。なお、この役割とAI事業者ガイドライン(5ページ)での分類との関係は、サービス開発者はAI開発者の一部とAI提供者、サービス提供者はAI利用者に対応されるとみなせる。

本文書で言及されるリスクは、2023年度までにデジタル庁で行われた行政でのテキスト生成AIの利活用の検証時に考察されたユースケースを元に想定されており、テキスト生成AI全般を網羅したものではない点に注意いただきたい。この検証については、[デジタル庁ホームページでの報告書](#)とその報告書を解説した[デジタル庁 Tech note](#)に詳細が書かれている。

本文書の利用は、テキスト生成AIの利活用を計画しているシステム管理者が、自身のユースケースに応じた想定リスクとその軽減策を参考に、テキスト生成AIに固有のリスクを考慮したリスクマネジメントの実施を想定している。

1.2 本文書で想定するテキスト生成 AI の利活用方法

本文書ではテキスト生成 AI を以下の 4 つのユースケースを想定している。

- ユースケース 1. チャットインターフェースでサービス利用者とインタラクティブに対話する機能としてテキスト生成 AI をオンライン処理で用いる
- ユースケース 2. 大量の文章に対してラベル付けやテキストデータ変換、その他翻訳や要約や文章作成等の自然言語処理を行う機能としてテキスト生成 AI をバッチ処理で用いる
- ユースケース 3. 情報検索を目的としたサービスで検索エンジンの補助としてテキスト生成 AI をオンライン処理で用いる
- ユースケース 4. ダッシュボードやソースコード等をサービス利用者の自然言語により記述可能にする機能としてテキスト生成 AI をオンライン処理で用いる

テキスト生成 AI の利活用でもっとも有名な [OpenAI 社の ChatGPT](#) はこの類型ではユースケース 1 に該当する。テキスト生成 AI の利活用はチャット利用に限定されない。

ユースケース 2 の一例は [2.2 2\) パブリックコメントの返答作成業務にテキスト生成 AI を適切に利活用するユースケース](#) で詳細に述べる。

ユースケース 3 の一例は [2.2 1\) 情報検索サービスを国民向けに展開する事例でテキスト生成 AI を適切に利活用する方法](#) で詳細に述べる。

ユースケース 4 の一例は [2.2 4\) 信頼できるデータベースと連携してテキスト生成 AI を適切に利活用するユースケース](#) で詳細に述べる。

1.3 本書の構成

本文書は実践ガイドブックをベースにし、そこに対してテキスト生成 AI 固有のリスクや留意点を補足することを目指している。そこで、「第 3 編第 1 章 実践ガイドブックの構成 Point.2 チェックリスト」の分類にならない、新サービス企画時、予算要求前、調達実施前、設計・開発時、サービス実施時の段階で検討すべきテキスト生成 AI 固有のリスクとその軽減策を述べる。

1.4 用語

用語	解説
生成 AI	出力が単一の数値やラベルではなく、文章や画像や音声や動画といった従来は人間が創作していた類のものを出力する機械学習モデルを使ったシステムのことを総称したもの。
テキスト生成 AI	生成 AI のうち、出力が文章に代表されるテキストのもの。出力は文章以外でもソースコード等やテキストデータの形式を変換したものといった自然言語以外の出力や、文章分類や固有表現抽出等といった従来の自然言語処理課題を行うことも可能である。
大規模言語モデル	<p>昨今のテキスト生成 AI の背後に多くのニューラルネットワーク型の機械学習モデルの一種で、膨大なパラメータ数と膨大なテキストにより学習されていることが特徴の一つ。</p> <p>テキスト生成 AI の背後は必ずしも大規模言語モデルではない（機械翻訳での LSTM 等）点や、大規模言語モデルの小型化研究が盛んであり必ずしも「大規模」とは限らない点が混乱しやすい。</p>
プロンプト	<p>大規模言語モデルへ入力するテキストデータのこと。</p> <p>膨大なテキストによる教師なし学習を事前学習とよび、特定の課題（下流タスクとも呼ぶ）に特化させた事後学習（ファインチューニング）とよぶ。以前はこの事後学習によりさまざまな自然言語処理課題に対応させる手法が主流だったが、大規模言語モデルではプロンプトによる指示を変えるだけでさまざまな自然言語処理課題に対応することが可能になった。</p>
トークン	自然言語を機械学習モデル等のシステムに扱いやすい単位で分割したもの。日本語の場合、検索エンジンへは n-gram や形態素といった単位でトークンが作られることが多く、大規模言語モデルへはサブワードといったさらに細かい分割が一般に行われる。

表2 本文書の用語集

2 新サービス企画時のテキスト生成AI固有の留意

本章では、テキスト生成AIの新たなサービス企画時における利活用の適切性を評価し、その結果生じうるリスクを特定し、これらのリスクに対する軽減策を提供することを目的とする。具体的には、不適切な利活用事例と効果的な利活用事例を識別し、政府情報システムにおけるテキスト生成AIの適用可能性を探る中で明らかにしていく。この章の範囲は、実際のサービス企画段階でのリスク評価と、それに基づいた意思決定支援に焦点を当てている。

2.1 テキスト生成AIの利活用が不適切であるケースにテキスト生成AIを用いるリスク

テキスト生成AIの利活用が不適切である場合にテキスト生成AIを用いることによりリスクが発生することが考えられる。例えば以下の6類型が考えられる。

- (1) テキスト生成AIによる回答の期待品質が高すぎる場合
- (2) そもそも人間が行うべき仕事であり、AIによる代替が不適切な場合
- (3) 特定の資格を必要とする作業であることが法令に定められている場合
- (4) テキスト生成AIの知識に無い事項を答えさせる場合
- (5) テキスト生成AI利活用の構築・運用費が費用対効果に見合わない場合
- (6) その他、AI事業者ガイドラインの「共通指針」(ページ12~20)に違反する場合

(1)は例えば、ユースケース1のチャットインターフェース型のサービスを国民向けに展開し、質問内容やサービスの期待値を特定のドメインに限定しない場合に発生しやすい(組織の見解と異なる発言や、差別的言動や虚偽の情報などの不適切発言を行う等)。チャットインターフェース上でAIが回答するサービスを行政が行う場合、規約等に十分記載したとしても、国民からの期待値を限定させることが難しいと考えられるため、国民向けに広くユースケース1を展開することは難しいと考えられる。

(2)は例えば、パブリックコメントの返答をテキスト生成AIで全自動化するケースは不適切な事例である。これはパブリックコメントの本来の目的は国民からの質問に回答することではなく、広く国民の声を行政に届けることが目的であるため、行政職員がパブリックコメントを読み込むことが本質であるからである。

(3)は例えば、症状についての記述から必要な薬をテキスト生成AIに推定さ

せ、薬の処方箋を自動発行するケースは不適切な事例である。医師法や歯科医師法に違反する。

(4)はテキスト生成 AI の学習データに無い情報を聞いた場合に確実に発生する。例えば、デジタル庁 AI 班の職員数をテキスト生成 AI に聞いた場合、根拠のない事実と異なる適当な人数を回答することがある。これはテキスト生成 AI の幻覚（ハルシネーション）と呼ばれる現象で、不適切な利活用例である。テキスト生成 AI の学習データに仮に情報があったとしても大規模言語モデルがそれを知識として確実に学習できるとは限らない点にも注意。

(5)はテキスト生成 AI の利活用形態にも依存する。**3 予算要求前時のテキスト生成 AI 固有の留意点**の箇所述べる。特に方法1のテキスト生成 AI をオンライン処理で用いる場合はリクエストあたりの運用費用が通常のシステムと比べて高くなる傾向がある。

(6)は具体的には AI 事業者ガイドラインの本文の他、別添の各部ごとの「「共通指針」の解説」を参照していただきたい。例えばアイデア出しを特定の大規模言語モデルに過度に依存した場合、得られるアイデアがその大規模言語モデルのバイアスに影響を受けてしまう危険性がある。ただし、この問題は大規模言語モデル固有ではなく、例えば内部職員や外部有識者の意見でも様々な種類のバイアスが存在するため、同様に注意が必要である。

これらの事例は、そのままの形ではテキスト生成 AI の利活用が適切になってしまいが適切な工夫を行うことでテキスト生成 AI を有用に扱うことができる。これを次節で述べる。

2.2 テキスト生成 AI の利活用が効果的なケースを見落とすリスク

前節の**2.1 テキスト生成 AI の利活用が適切であるケースにテキスト生成 AI を用いるリスク**で挙げた(1)(2)(3)(4)の事例では工夫次第ではテキスト生成 AI 利活用により政府情報システムや業務を改善することができる。

1) 情報検索サービスを国民向けに展開する事例でテキスト生成 AI を適切に利活用するユースケース

この事例では、まずサービス利用者に本当に提供したいものがユースケース1のチャットインターフェースなのか、ユースケース3の情報検索を目的としたサービスに生成 AI を活用したものなのかの検討を行う。おそらくほとんどの場合は国民向けに利便性の高い情報検索サービスを提供したい、つまりユースケース3になるだろう。

ここでは、情報検索サービスにテキスト生成 AI を利活用する方法の一例を、旧来の利便性のあまり高くないチャットボット型のサービスと比較して述べる。

旧来の利便性のあまり高くないチャットボット型サービスの典型的な挙動は以下である。

- チャットボットが潜在的に返すことのできる回答候補の数が少なく、必要とする回答をサービス利用者がどんなに入力しても得られない
- サービス利用者に文章で入力させるが、文中の単語の一部を抽出するだけで文意は無視される。また単語で検索されるが単語の表記揺れなどは考慮されない。
- チャットボットから返された検索結果の内容の理解に困難を伴い、関連が低い回答内容を読解するためのサービス利用者への負荷が高い。

これらの課題は情報検索システムを中心とすると下記のように図示できる。



図 1 情報検索システムでユーザビリティ向上を生成 AI で実現できる箇所

これらすべての箇所でテキスト生成 AI は旧来のチャットボット型サービスあるいは検索システムを改善できる。詳細は [8 従来型の情報検索サービスをテキスト生成 AI により改善する手法](#) で述べる。

テキスト生成 AI による国民向けへの質の高い情報検索サービスの構築には、いわゆる ChatGPT のようなユースケース 1 に限らず、ユースケース 3 の旧来の情報検索システムの考え方を前提にしたテキスト生成 AI による利便性の向上にも検討の価値がある。特にこの手法の優位点は事前に準備した回答候補をベースとするため原理的には事前にすべての回答パターンをテストできるため高い品質要件を満たすことができる点である。また、テキスト生成 AI を主にバッチ処理で用いることでコスト予測が立てやすいため予算要求前での見積もり精度を上げることもできる。こちらは [5. 2 4\) テスト済みの生成物のみを用いる場合の工夫](#) でも詳細に述べる。

2) パブリックコメントの返答作成業務にテキスト生成 AI を適切に利活用するユースケース

パブリックコメントの返答作成業務は行政職員が各パブリックコメントを読み込むことにその意義がある。しかしパブリックコメントが大量に届いた場合に、限られた職員数で限られた時間内にパブリックコメント内に含まれる意見を全て拾い上げることが現実的に厳しい場合が存在する。その際に例えば以下のような方法でテキスト生成 AI を用いることで、職員によるパブリックコメントからの意見抽出の品質を向上できる可能性が高い。

- (1) 職員がパブリックコメントの全量を確認し、意見内容の類型化を行う
- (2) ひとつのコメントに複数の内容が含まれることも多いため、テキスト生成 AI を用いてコメントを内容単位で分割する
- (3) 分割したコメント内容をテキスト生成 AI により類型化したラベル付けを行う
- (4) ラベルごとに職員が妥当なラベル付けであるかを確認する
- (5) 類型化したラベルがつけられなかったコメントは見落としした意見である可能性が高いため職員が改めて精読する

(5)の工程があることで(1)の作業に過度の無謬性を必ずしも求めなくて良くなるため職員の精神的負担も軽減されると考えられる。また(4)の工程は職員が行う手間が純増する箇所ではあるが、文のラベル付の妥当性判断は文から自身でラベル付するよりもだいぶ負担が少ないため相対的に時間をかけずに行うことができる。

3) 法令で決められた資格保有者等の支援についてテキスト生成 AI を適切に利活用するユースケース

薬の処方箋と医師のように、業務の中には資格をもった従事者が必須とされるものがある。この場合、医師をテキスト生成 AI で代替することは不可能であるが、医師をテキスト生成 AI でサポートする方法は考えられる。例えば、薬の処方箋の原案を何パターンかテキスト生成 AI が医師へ提案し、医師自身の中で妥当なものがあれば一番良いものを選ぶ、といった形で医師の業務支援を行うといった方法が考えられる。

これはあくまでわかりやすさを重視した一例であって、実際には資格保有者の業務をちゃんと理解した上で必要な支援を検討することが必要である。業務の現状把握の重要性については実践ガイドブックでも述べられている

4) 信頼できるデータベースと連携してテキスト生成 AI を適切に利活用す

るユースケース

不適切な例に挙げた「デジタル庁 AI 班の職員数」についての回答は、デジタル庁の人事情報が含まれるデータベースに対して適切なデータベースへの問い合わせ（クエリ）文を発行することで正確な数字が得られる。このクエリ発行にテキスト生成 AI を利活用することで、自然な文章での入力から正確な回答が得られるシステムを構築することができる。このような生成 AI の活用方法は、ダッシュボードに組み込まれた製品として利活用することも可能である。製品利用の場合も、特別に開発する場合も、データベースのテーブルやカラムの定義をテキスト生成 AI に与えないと正確なクエリ文生成ができない点には注意が必要である。

2.3 技術的実現可能性の検討によるリスク軽減

前節の **2.2 テキスト生成 AI の利活用が効果的なケースを見落とすリスク** で挙げた活用事例はあくまで机上の話であり、実際に技術的に実現可能かは個別具体のユースケースやデータに依存するため確かなことは言えない。最悪の場合、調達後に技術的実現可能性がなかったことが判明するリスクもありうる。

テキスト生成 AI の重要な特徴のひとつにプロンプトの変更だけで多様なユースケースに対応できるというものがある。この特徴を使えば AI の専門家でなくても、テキスト生成 AI でどのようなことができそうかを事業者に頼らずに検討することが可能である。

例えば、情報検索システムにテキスト生成 AI を組み込む場合は、自然文からの意図推定を指定したフォーマットで十分な精度で実施できそうかを下記のようなプロンプトで検討できる。

テキスト生成 AI への入力例

下記の「ネコに国政選挙の選挙権を与える政策」に対するコメントから「賛成か否 (is_agree)」「提言(suggestion)」を json 形式で抽出する

コメント

子どもに選挙権がないのに猫に選挙権を与えることはいかがなものかと思う。せめて子猫には選挙権与えないといった猫の中での対応が必要

output format

```
{  
  "is_agree": true|false,  
  "suggestion": "{#text}"  
}
```

テキスト生成 AI からの出力例

```
{  
  "is_agree": false,  
  "suggestion": "猫の中での対応が必要"  
}
```

図2 テキスト生成 AI で出力形式を json 形式に指定した例。この例での生成物の json の内の suggstion だけを最終的な出力としてサービス利用者に戻すことで、プロンプトインジェクションにもより頑健になる（絶対ではない点に注意）。本例では Swallow-70b-instruct-hf を利用した。

ラベリング課題でどのくらいの精度が出せそうかについては下記のようなプロンプトで検討することができる。

テキスト生成 AI への入力例

下記のコメントが「ネコに国政選挙の選挙権を与える政策」に対して賛成か反対かを判定せよ

コメント

子どもに選挙権がないのに猫に選挙権を与えることはいかがなものかと思う。せめて子猫には選挙権与えないといった猫の中での対応が必要

output format

{賛成|反対}

テキスト生成 AI からの出力例

反対

図3 テキスト生成 AI でラベリング課題を実施させた例。本例では [Swallow-70b-instruct-hf](#) を利用した。

候補出し課題でどのくらいの精度が出せそうかについては下記のようなプロンプトで検討することができる。

テキスト生成 AI への入力例
「ネコに国政選挙の選挙権を与える政策」に対する賛成意見と反対意見を3つずつ作って
テキスト生成 AI からの出力例
はい!「ネコに国政選挙の選挙権を与える政策」に対する賛成意見と反対意見を3つずつ作りました。
●賛成意見
・猫が国政選挙に参加できるようになることで、多様な意見や価値観が反映され、より公正かつ民主的な政治システムが実現される可能性がある。
・猫が国政選挙に参加することで、人間だけでなく動物にも配慮した社会づくりが進む可能性がある。
・猫が国政選挙に参加することで、政治への関心が高まり、投票率が上がる可能性がある。
●反対意見
・猫が国政選挙に参加することで、人間と同じ権利を持つことになり、法律上の問題が生じる可能性がある。
・猫が国政選挙に参加することで、猫の福祉や健康面に悪影響を及ぼす可能性がある。
・猫が国政選挙に参加することで、猫に対する偏見や差別が助長される可能性がある。

図4 テキスト生成 AI で文章を作成させた例。本例では [Swallow-70b-instruct-hf](#) を利用した。

これらのプロンプトは非常に簡素なものだが、今回の課題がどのくらい難しいものなのかについては実際の想定データや出力形式をプロンプトに入れた検討ができることを示している。

この時点である程度、事業者ガイドラインの「共通指針」の中の「人間中心」の要件を満たせそうか、どのように確認すべきか、どこまでを目標にするかについて検討も行えると良い。具体的手法は別添(ページ130等)にも記載されているが、より具体的なテストケースをどのようにどこまで作るべきか、はプロジェクトの性質にも依存するため一概に決定することは困難である。傾向として、テキスト生成 AI にもたせる役割を明確にした方が必要なテストケース数を

削減でき、テスト観点も明確にできる。特にユースケース 1 では(1)テキスト生成 AI が回答するテーマを明確にする(2)そのテーマ以外では回答させない、といった工夫を行わないと、テストケースが絞り込めず莫大になってしまい、開発したシステムが十分な品質かの確認が非現実的になってしまう。

実際に十分な品質要件が達成できるかについては開発時のテストで確認する必要がある。詳細は [5 設計・開発時のテキスト生成 AI 固有の留意点](#) の節で述べる。

大規模言語モデルの種類によっては課題によって精度が異なることが想定される。さらに一部の大規模言語モデルでしか検討できないことにより、次節で述べる提供形態の選択肢の自由度が狭まりベンダーロックインのリスクも高まるため、職員が複数の大規模言語モデルに対して検討できる状態が望ましい。大規模言語モデルは既存のテキストデータだけから学習させるだけでなく、「どのような回答が望ましいか」について人間がラベル付けしたデータに基づいた追加学習が行われていることが多い。この「回答の望ましさ」については基盤モデル開発者やサービス開発者の方で決められており、この「望ましさ」の違いが、ユースケースによっては大規模言語モデルの選定の考慮事項になる。

(参考: OpenAI 社が公開している「望ましさ」について。 [How should AI systems behave, and who should decide?](#))

多くの大規模言語モデルではいくつかの制限はあるものの、無料で試用できる Web サービスが用意されていることがあり、それを用いての検討を行えば、期待品質を満たさない手戻りリスクを軽減することができる。

2.4 民間サービスの活用

- (1) テキスト生成 AI を利活用するための提供形態は大きく以下の 3 つに類型できる。
- (2) テキスト生成 AI を組み込んだサービスとして提供されるケース
- (3) テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケース

テキスト生成 AI の機械学習モデルが直接提供されるケース

(1) は SaaS サービスとして提供されることが多い。(3) はクラウドサービスやオンプレミスでサーバを調達して機械学習モデルをデプロイして使うケースを想定している。「自分で作りすぎない」「クラウド・バイ・デフォルトの原則」から(1)、(2)、(3)の順番での検討を推奨する。必須要件へのカスタマイズ性や取り扱う情報の機密性の高さから(1)や(2)よりも(3)を合理的に選択するケースも十分にありうるため、企画の内容に応じて精査を推奨する。この選択に利用できる実践的なフレームワークやチェックリストは現時点(2024年6月現在)

で開発できていないため、ここでは留意点の紹介に留まる。

提供形態	開発コスト	カスタマイズ性	運用コスト
テキスト生成 AI を組み込んだサービスとして提供されるケース	<p>既にあるものを利用するため基本的には無い</p> <p>既存の情報システムにテキスト生成 AI を組み込む場合は選択肢に挙がらない</p>	<p>基本的には低い。サービス提供元会社と直接委託契約を結び、こちらの要望を積極的に取り入れる場合は、カスタマイズ性も高くなる</p>	<p>サービス利用料程度しかかからず、既に利用マニュアルやヘルプデスクが準備されている場合もあり</p>
テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケース	<p>Web API を経由でテキスト生成 AI を利用するだけであれば開発コストは少ない</p> <p>既存の情報システムにテキスト生成 AI を組み込む場合はプロンプトの試行錯誤やテスト工数が通常のシステム開発と比べて大きい</p>	<p>さまざまな種類の大規模言語モデルが主要なクラウドサービスによって Web API 形式で提供されている。独自に学習した大規模言語モデルを使うオプションもあり、テキスト生成 AI 部分のカスタマイズ性は最も高い</p>	<p>入出力のトークン数や文字数に比例する料金形態をとっていることが多く、コスト試算が独特になる。利用する大規模言語モデルによって価格が大きく異なるため、品質要求と運用コストのバランスが求められる</p>
テキスト生成 AI の機械学習モデルが直接提供されるケース	<p>オンプレミスでのサーバ調達やクラウド上での計算環境を用意し、その上にテキスト生成 AI を利用するための開発が必要になるため、開発コストは最も高い</p>	<p>利用できる大規模モデルに制約がかかり、最高峰の大規模言語モデルが利用できない可能性が高い</p> <p>インターネットを利用しなくても良いので、ネットワーク要件やセキュリティ要件に応じたカスタマイズ性は最も高い</p>	<p>ハードウェアのメンテナンスやメモリ管理などでやらなければならないことが多いため、3 つの中では最も運用コストが高い傾向がある</p> <p>ただし潤沢な計算リソースが確保でき、テキスト生成 AI の利用頻度が非常に多い場合に、最も運用コストが抑えられる可能性もゼロでは無い。</p> <p>大規模言語モデルの小型化も盛んに研究されている</p>

			るため、将来的には現実的なコストで収まる可能性は高い
--	--	--	----------------------------

表3 テキスト生成 AI の提供形態による違いを簡単にまとめたもの

提供形態ごとに考慮すべきリスクを、ライセンス、調達容易性、機密性、アップデート対応、ベンダーロックイン、コストマネジメントの 6 観点で整理したものを付録の 7 提供形態ごとの想定リスクにまとめたため、そちらも参照いただきたい。

2.5 2章のまとめ

新サービス企画時には、テキスト生成 AI の利活用が不適切であるケースと効果的なケースを見極めることが重要である。本章では、適切な工夫によってリスクを軽減し、テキスト生成 AI を有効活用する例を示した。実際のユースケースに応じた想定リスクとその軽減策が、サービスの質を高め、不測の問題を未然に防ぐための知見となることを目指している。

3 予算要求前時のテキスト生成AI固有の留意点

予算要求前におけるテキスト生成AIのリスクと機会の評価を目的とした本章では、異なる提供形態に応じた予算策定のための考慮点を提供する。予算計画の精度を高めるために、コスト削減の観点からの実用的な留意点と、適切な提供形態を選択するための留意点を挙げる。スコープは、テキスト生成AIを活用する際の財務的なリスク管理と、資源配分の最適化に重点を置いている。

テキスト生成AIを利用するための提供形態の違いにより、必要な予算の金額が大きく異なるため、予算要求前に提供形態の目途をつけるべきである。適切な提供形態の選択に失敗すると、予算の大幅な超過あるいはその逆が発生するリスクが大きい。このリスクは各利用形態の特徴を把握し、利用形態ごとに適切なコスト削減に向けた検討を行うことによって軽減できる。ここでは、提供形態ごとに留意すべき点を述べる。

どのケースでも事業者ガイドラインと別添の「第2部 AIにより目指すべき社会及び各主体が取り組む事項」と「第5部 AI利用者に関する事項」を参照していただきたい。テキスト生成AIを利用するためのWeb APIがクラウドサービスとして提供されるケースとテキスト生成AIの機械学習モデルが直接提供されるケースの場合は、「第4部 AI提供者に関する事項」と、必要に応じて「第3部 AI開発者に関する事項」も参照していただきたい

3.1 コスト削減に向けた検討

1) ハードウェア・ソフトウェアのコスト削減観点

ア テキスト生成AIを組み込んだサービスとして提供されるケース

類似サービスの価格を調べ、市場の相場と比べて高価すぎないかの確認を行う。

イ テキスト生成AIを利用するためのWeb APIがクラウドサービスとして提供されるケース

性能が良いとされる大規模言語モデルほど、リクエストあたりの費用が高くなる傾向にある。また、リソースが共有される汎用的な基盤モデルの利用よりも、再学習を行った独自の大規模モデルの利用の方が、費用が高くなる場合が多い。テキスト生成AIの生成物の品質目標を定め、事前にどのくらいの性能の大規模言語モデルで十分かの検討を行う。ただし、実際に開発が進むにつれて

当初想定した大規模言語モデルでは性能不十分であることが発覚するリスクも十分考えられる点にも注意する。

また、回答精度を上げるためにプロンプトへの記述を手厚くするほど、トークン数や入力文字数に比例し費用が高くなるため、プロンプトの改良による精度改善の試行錯誤により初期検討時のプロンプトよりも費用が高くなる可能性も十分考えられる。想定よりもリクエスト数が多い場合に想定以上の費用が高くなる可能性もある。

この場合、初期検討時のものと比べてどこまで費用が上振れするものかのデータがまだないため、妥当な見積もり方法が確立していない点に注意が必要。

クラウドサービスによっては従量課金でなく定額プランを用意している場合もあるが、定額プランは一般的にかなり高価である。しっかりと見積もりをしたうえで採用することを強く推奨する。

ウ テキスト生成 AI の機械学習モデルが直接提供されるケース

多くのケースで最もコストが高い選択肢になる傾向がある。少なくともハードウェアの調達から必要な場合は、初期投資が最も高い。他の利用形態で要件を十分満たすかの検討とコスト試算を行ったうえで、この利用形態を採用するかを検討する。

2) アプリケーションのコスト削減観点

ア テキスト生成 AI を組み込んだサービスとして提供されるケース

テキスト生成 AI で扱われる大規模言語モデルは、ひとつのモデルだけで多様な自然言語処理（例えば、翻訳や要約、文章作成や固有表現抽出等）が可能である。この特徴から、バラバラに実装・提供されていた複数の自然言語処理関連の機能や製品をまとめられないかの検討をし、重複投資を避ける。

イ テキスト生成 AI を利活用するための Web API がクラウドサービスとして提供されるケース

生成 AI へのリクエスト 1 件あたりの平均費用を下げるか、そもそもリクエスト量を減らすことでコスト削減が可能である。平均費用の方は **3. 1 1)** **ハードウェア・ソフトウェアのコスト削減観点** で述べたため、ここでは生成 AI へのリクエスト量を減らすための検討事項を紹介する。

リクエスト量を大幅に減らすことのできる選択肢として主なものは以下の 3 つである。

a) 複数のリクエストを一つに集約する

テキスト生成 AI の Web API への複数リクエストを一つに集約すればコストカットにつながる可能性がある。ただしこの手法ではリクエスト 1 件あたりのトークン数が多くなるため、リクエスト 1 件あたりの平均費用が高くなり、結果としてコストカット効果に乏しい場合もある。また、テキスト生成 AI にとっての処理難易度が上がることで品質劣化を招く危険性もある。

b) テキスト生成 AI 以外の手法を採用する

例えば、文中の単語や特定の品詞を抽出するだけであれば形態素解析器で十分であり、文間の意味的類似度を測るだけであればベクトル表現（埋め込み表現、embedding ともいう）の採用で十分な場合もある。このようにテキスト生成 AI 以外の手法で十分な場合ではテキスト生成 AI 以外の手法をまず検討すべきである。またこれらの自然言語処理手法は、テキスト生成 AI との二者択一ではなく、相互に弱みを補完しながら組み合わせて利用することも十分に考えられる。

c) 過去のテキスト生成 AI の生成物を再利用する

過去にテキスト生成 AI で生成された実績のあるリクエストと同じようなリクエスト（例えばリクエスト文字列をベクトル表現化し、それと近いベクトル表現の場合に、同じようなものとみなす）の場合、キャッシュサーバー等から過去の生成物を送信するようにし、テキスト生成 AI の Web API にリクエストをさせない。この手法をより極端にした場合、テキスト生成 AI の Web API へのアクセスは全て事前にバッチ処理で行い、テキスト生成 AI 関連の処理にかかるコストを制御しやすくする手法も考えられる。この手法に関しては、[5. 2 4\) テスト済みの生成物のみを用いる場合の工夫](#)で詳細に述べる

ウ テキスト生成 AI の機械学習モデルが直接提供されるケース

大規模言語モデルの実行にともなう計算コストに直接影響される形でハードウェアや運用に関連するコストが増加する。そのため大規模言語モデルの実行にともなう計算コストが重要であり、その観点や手法は Web API 利用のケースでの検討事項と同様である。

3) 運用業務のコスト削減観点

ア テキスト生成 AI を組み込んだサービスとして提供されるケース

アカウント管理に関連する業務コストは無視できないほど大きい。これは新

規着任者だけではなく、異動者や退職者のアカウント管理も必要となり、組織変更と同期が求められるからである。既に組織に紐づいた形でのアカウント管理が導入されているならば、そのアカウントでの認証に用いられるシングルサインオンでアカウント管理の導入ができれば、アカウント管理業務のコストは大幅に削減できる。例えば、ガバメントソリューションサービス（GSS）の Microsoft Entra ID のシングルサインオンがそれに該当する。

イ テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケース

一般的なシステム開発と大きな違いはないが、テキスト生成 AI の Web API のキャパシティプランニングが入出力のトークン数にも依存するといった特殊性や標準的な Web API とくらべてレスポンスタイムが遅い（ただし、高速化する方向での開発も盛んである）傾向にある。そのため、これらの指標についてのサービスレベルや監視の設定に注意が必要である。テキスト生成 AI の Web API への負荷を下げることであれば、これらの問題も軽減できる。この負荷を下げる手法は **3. 1 2) アプリケーションのコスト削減観点** を参考にしていきたい。

ウ テキスト生成 AI の機械学習モデルが直接提供されるケース

一般的なシステム開発と大きな違いはないが、GPU メモリが大規模言語モデルの出力トークン数の影響を強く受けるといったキャパシティプランニングに特殊性がある。そのため、これらの指標についてのサービスレベルや監視の設定にも注意が必要である。この負荷を下げる手法は **3. 1 2) アプリケーションのコスト削減観点** を参考にしていきたい。

4) その他のコスト削減観点

ア テキスト生成 AI を組み込んだサービスとして提供されるケース

アカウント数に比例した価格体系であることも多く、本当に必要とするサービス利用者のみアカウントを配布する。常駐のヘルプデスクではなくマニュアルや動画コンテンツでの解説等を採用する。ただしテキスト生成 AI を用いた業務改善プロジェクトのような不確実性の高いプロジェクトの場合、本当に必要とするかの見極めは難しく、潜在的な業務改善効果は高いがテキスト生成 AI に不慣れなサービス利用者のために手厚いサポートが重要であることも考えられるため、一般的な SaaS 製品の導入と比べて検討が困難な傾向がある。

イ テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケース

すでにテキスト生成 AI を利用したシステムがある場合、実績をベースにコスト削減が可能かの検討を行う。例えば、定額プランを採用していた場合、実際のアクセス数から従量課金に見直せないかの検討を行う等が考えられる。

ウ テキスト生成 AI の機械学習モデルが直接提供されるケース

すでにテキスト生成 AI を利用したシステムがある場合、実績をベースにコスト削減が可能かの検討を行う。例えば、機微な用途が限定されていたことからセキュリティ要件を見直し、Web API 形式の場合のコスト検討を行う等が考えられる。

3.2 3章 まとめ

予算要求前時には、選択された提供形態に応じて大幅に異なる予算規模を見積もる必要がある。本章では、コスト削減を目指した各提供形態の留意点を詳述し、AI 事業者ガイドラインとの観点にマッピングしながら、実用的なリスク軽減策を紹介しました。これらの対策が予算超過や不足を防ぎ、効率的なプロジェクト運営をサポートする一助となれば幸いである。

4 調達実施前時のテキスト生成AI固有の留意点

本章の目的は、調達実施前の段階で、テキスト生成 AI に関連する非機能要件を明確にし、サービス実施時の品質保証を確立することである。テキスト生成 AI の利活用を計画しているシステム管理者が、サービスの期待品質を具体的に理解し、これを適切に盛り込んだ調達要件を作成できるようにすることをスコープとする。

非機能要件の明確化をすることで、サービス実施時に問題が発覚することによるシステムが一般的に期待される品質（非機能要件に該当しやすい）を満たさなかった場合におきる炎上リスクや開発手戻りリスクを軽減する。通常システム開発の非機能要件と比べて、テキスト生成 AI を用いたシステムの場合、テキスト生成 AI の生成物に対するサービスの期待品質や、AI 事業者ガイドラインで挙げられている「人間中心」、「安全性」、「公平性」、「プライバシー保護」、「セキュリティ確保」、「透明性」、「教育・リテラシー」、「公正競争確保」、「イノベーション」の指針を考慮する。

AI 事業者ガイドラインでの指針では別添に具体的な手法が掲載されているため、本章ではサービスの期待品質についての観点をユースケースごとに具体的に考察を行う。

4.1 チャットでの利用の期待品質

ここではユースケースのうち、チャットインターフェースで利用者サービス利用者とインタラクティブに対話する機能としてテキスト生成 AI をオンライン処理で用いるケースでの期待品質について述べる。理想的な応答のテストケースの量と質についての要件を決める。例えば、このシステムがゴミ出し方法のルールについてのものであれば、ゴミの種類や捨て方の方法ごとに 10 種類、全く関係ない質問を 100 種類作成し、その質問の 99%以上で妥当な回答を返すことを要件に入れる。妥当かどうかの基準はできるかぎり明確にする。この基準の明確化については **5.2 2) 生成物の品質評価が困難であることに起因するリスク** で詳細にふれる。

この期待品質の基準が現実的なものか、は問題の難しさにも依存するため、一概に決め切るのが難しく、その基準が妥当なものかの判断も非常に難しい。しかし、期待品質を明確にしないと、期待品質を下回るシステムが納品されるリスクがある。軽減策として、この種のシステムを委託開発する場合は、プロポーザル型の調達方法を採用し、調達要件を作成するさいに複数の事業者に相談することを推奨する。そのさいに事業者から回答された現実的な品質レベル

が受け入れにくいものであれば、まだ求めることに対して技術が追いついていないとみなし、プロジェクトの中止を考慮に入れた検討を推奨する。

4.2 バッチ処理での利用の期待品質

ここではユースケースのうち、大量の文章に対してラベル付けやテキストデータ変換、その他翻訳や要約や文章作成等の自然言語処理を行う機能としてテキスト生成 AI をバッチ処理で用いるケースでの期待品質について述べる。テストケースを充実させることで期待品質を満たすかの確認を要件に入れ、低品質なシステム納品リスクの軽減ができる。このケースでは偽陽性・偽陰性のバランスを考慮し、実用的な受け入れ基準を決めることが重要である。

例えば、書類審査業務の代わりにテキスト生成 AI で行う場合を考える。まずリスクを「本来は審査 NG とすべき書類を審査 OK としてしまう（偽陽性）」「本来は審査 OK とすべき書類を審査 NG としてしまう（偽陰性）」の2つに分けて整理する。それぞれでどのくらいのミスが許容できるかを検討する。

「本来は審査 NG とすべき書類を審査 OK としてしまう（偽陽性）」がそのままでは全く許容できない場合、テキスト生成 AI で審査 OK となった場合は必ず人間でも審査することで軽減できる。この場合は(1)テキスト生成 AI に審査 NG と判定されたケースを人間が審査しないことによる業務削減や、(2)審査 OK にした理由をテキスト生成 AI に作成させることによる審査過程の補助による改善効果が見込まれる。

「本来は審査 OK とすべき書類を審査 NG としてしまう（偽陰性）」がそのままでは許容できない場合、審査 NG 理由をテキスト生成 AI で生成させ、その結果を申請者にフィードバックし、(1)修正し再提出してもらう(2)理由に納得いかない場合に備え、テキスト生成 AI を介さず直接人間が審査する口を用意する、といった方法をとればリスクが許容できるまで軽減できる可能性が高い。

これらの方法を採用することでは人間による審査をゼロにはできず、テキスト生成 AI による業務改善効果が限定化されてしまうが、しかし、全てを人間で実施する場合と比べて、審査リードタイム（申請側にとってメリット）や審査業務の負荷（審査側にとってメリット）の改善が期待できる。

また、ラベリングした結果が多少不正確でも許容できるユースケースの場合は、サービス利用者からのフィードバックを反映する仕組みを導入する等の運用を始めてから改善していく、と割り切った考え方もできる。

4.3 情報検索での利用の期待品質

ここではユースケースのうち、情報検索を目的としたサービスで検索エンジンの補助としてテキスト生成 AI をオンライン処理で用いるケースでの期待品質について述べる。テストケースを充実させることで期待品質を満たすかの確認を要件に入れ、低品質なシステム納品リスクの軽減ができる。また、テキスト生成 AI をオンライン処理で用いることで、レスポンスタイムが大幅に悪化するため、レスポンスタイムのサービスレベルの検討も必要になる。通常の情報検索サービスではレスポンスタイムのサービスレベルが数秒以下に設定されることが多いが、テキスト生成 AI をオンライン処理で用いた場合にこのサービスレベルが達成できない可能性がある。レスポンスタイムの悪化が許容できない場合、テキスト生成 AI はバッチ処理のみの利用（5.2.4 テスト済みの生成物のみを用いる場合の工夫も参照いただきたい）やテキスト生成 AI 以外の手法をうまく組み合わせる（8 従来型の情報検索サービスをテキスト生成 AI により改善する手法も参照いただきたい）ことを検討する。

期待品質を満たすかの基準としては、一般的な情報検索サービスと同様に、再現率と適合率をわかりやすい受け入れ基準にすることができる。

4.4 自然言語からのプログラム作成での利用の期待品質

ここではユースケースのうち、ダッシュボードやソースコード等をサービス利用者の自然言語により記述可能にする機能としてテキスト生成 AI をオンライン処理で用いるケースについての期待品質を述べる。これもテストケースの充実をさせることで期待品質を満たすかの確認を要件に入れ、低品質なシステム納品リスクの軽減ができる。ダッシュボードの数値だしの場合、問い合わせ文章と人間が実施した場合に得られる正解の数字の対を何パターンか用意し、どのくらいの正答率を基準にするか、で受け入れ基準を作成できる。例えば、性別・月ごとの区市町村の人口のデータベースに接続したダッシュボードの場合、「2024年1月の東京都の男性人口は？」という問いに対し、あらかじめ集計した「6918630」という回答を用意し、この数字と同じものを出力できるか否か、をテストする。

4.5 4章 まとめ

調達実施前には、非機能要件を明確化し、サービス実施時の品質を保証することが不可欠である。本章では、ユースケースごとの期待品質を定め、プロジ

プロジェクトの品質目標を達成するための具体的な検討を示した。この段階での正確な要件定義が、プロジェクトの成功において決定的な役割を果たすため、本検討例が参考になれば幸いである。

5 設計・開発時のテキスト生成AI固有の留意点

本章では、テキスト生成AIの設計・開発時に特有のリスク及び品質評価の課題に着目する。テキスト生成AI固有の留意点として、生成物の品質向上と品質評価が困難である点が挙げられるからである。そこで、本章の目的は、テキスト生成AIを用いたシステムの開発において、生成物の品質を保証し、その評価に関するリスクを軽減するための方法論を提供することである。スコープには、生成物の品質向上のためのテスト戦略と、その品質評価プロセスの提案を含んでいる。

設計時に留意すべき点は [3. 1 コスト削減に向けた検討](#)と [4 調達実施前時のテキスト生成AI固有の留意点](#)と重複するため割愛する。

5.1 テキスト生成AIの生成物の品質面への懸念

テキスト生成AIの生成物の品質を完全に保証することは困難である。ここでは実際の生成物が十分な品質でなかった場合のリスクとその軽減策について述べる。リスクの発生を予防する軽減策は、[5.2 テキスト生成AIによる生成物の品質評価・品質保証についてのリスクとその軽減策](#)も参照にしていきたい。

1) テキスト生成AIを分類やラベリングに利活用するケース

テキスト生成AIの生成物を文章として扱うのではなく、テキスト生成AIで分類やラベリング課題を解かせて、その結果を用いるケースで、その結果が不適切であった場合の軽減策について述べる。

この問題は、テキスト生成AI固有ではなく、従来のAI利活用の分類問題や人間参加型AI (Human In The Loop)で十分議論されてきた話題ではあるが、テキスト生成AI利活用のリスクを考える上で特に重要であるので、本文書で取り上げる。

軽減策

分類やラベリングの場合、どのような間違いがどこまで許容できるかを明確にした上で利活用を検討する。

[4.2 バッチ処理での利用の期待品質](#)の箇所も参照にしていきたい。

2) テキスト生成AIの生成物を文章として扱うケース

テキスト生成AIは分類やラベリングといった従来のAIと違って、自然な文章を生成する点に、その特徴がある。このリスクは(1)どのような文章を(2)誰

が(3)どのような目的で利活用するか、に応じて検討が必要なため、この章ではリスクそのものの説明は行わず、その内容は4 [調達実施前時のテキスト生成 AI 固有の留意点](#)で紹介する。ただし、これらリスクの軽減策には共通するところが多い。

軽減策

(1)生成物の出力形式を指定する

テキスト AI の生成物の出力形式を指定することで、以下の3種類のメリットがある

- (1) 出力される文章の内容が期待するものに近づく
- (2) 推論の過程を誘導する出力形式を採用することで生成物の品質が良くなる
- (3) 何らかの理由で出力形式通りでない結果が生成された場合、テキスト生成 AI 以外のプログラムで行われる単一の if 文レベルの簡単な処理によりエラー判定ができる

(2) 事前に十分テストする

テキスト生成 AI の利活用では十分なテストとは何か、から固有の考え方が必要となるため5.2 [テキスト生成 AI による生成物の品質評価・品質保証についてのリスクとその軽減策](#)で詳細に述べる。

5.2 テキスト生成 AI による生成物の品質評価・品質保証についてのリスクとその軽減策

テキスト生成 AI の品質に関わるテストが十分に行われないことにより、プロジェクトの品質目標を満たさないシステムをリリースしてしまうリスクが発生しやすくなる。テストが不十分であること自体のリスクはテキスト生成 AI 固有でなく情報システム全般で共通するが、テキスト生成 AI による生成物の品質をどのように評価・保証するか、についてはテキスト生成 AI 固有と考えられるリスクがいくつか存在する。本文書では以下の3つのリスクについて考察を行う。

- (1) 生成結果がばらつくことに起因するリスク
- (2) 品質評価が困難であるリスク
- (3) テストケースのカバレッジが不十分なことに起因するリスク

これらのリスクは、テキスト生成 AI の生成物はテスト済みのものだけ用いる方策によって大きく軽減するが、その方策を有用に用いるためにはいくつか制約がある。その制約について5.2 [4\) テスト済みの生成物のみを用いる場合の工夫](#)で考察する。

1) 生成結果がばらつくことに起因するリスク

テキスト生成 AI の機械学習モデルの推論過程には一般にばらつきが生じることがある。そのため同じ入力値であっても、対応する生成物が毎回同じ結果にならず、ばらつきまで考慮した評価が必要になる。が起こりうる。結果の制御困難なばらつきはテキスト生成 AI ではない一部の AI システムでも起こりうるが、大半のテキスト生成 AI で生成物のばらつきが見られることとその影響の大きさから、本文書で取り上げる。

想定リスク

事前にテストした結果と異なる結果を返すリスクが生じやすい。具体的にどのようなリスクが発生するかはユースケースに依存する。ここでは、ユースケースにあまり依存しない一般的な軽減策を述べる。

軽減策

(1) ばらつきに関するパラメータを調整する

ばらつきに関するパラメータを調整し、ばらつきを抑制する。この方法では、同じ入力に対して毎回同じ結果になる確率は上がるが、利用するサービスや機械学習モデルによっては、常に同じ結果になることは保証されないことも多く、注意が必要である。

(2) 生成物の出力形式を指定する

生成物の出力形式を指定することで、ばらつきを軽減することができる。
5.1.2 テキスト生成 AI の生成物を文章として扱うケースの(1)と同じ手法なのでそちらを参照いただきたい。

(3) 同じテストを繰り返し行う

品質評価時に同じ入力値で何回かテストし、生成物のばらつきが許容できる範囲か、ばらついた場合でも常にテストをパスできるかの検討を行う。

これは、生成物にばらつきが存在することを前提とした上で、テストの方法でそのリスクを軽減する方法である。

実際に何回テストするか等の詳細はユースケースによって異なる品質要件や、どのような入力に対してどのような出力を求めるかによって異なる。テキスト生成では生成結果に長文や創造性が求められるケースでばらつきが生じやすい。

(4) 同じ入力に対し同じ出力を返す仕組みを、テキスト生成 AI モデルの外側で実装する

どうしても毎回同一の結果でなければならない時は、推論時に入力と生成物

のペアを保存しておき、過去に同じ入力であった場合にはテキスト生成 AI を利用せずに当時の生成物を返す等といった外側のアプリケーション側の工夫で対策を実施する。テキスト生成 AI サービスの中にはその種の機能が備わっているものもあるため、自前で実装する前に、その機能や仕様について確認することを推奨する。

2) 生成物の品質評価が困難であることに起因するリスク

ア 一般的な想定リスク

テキスト生成 AI の成果物は、数値やラベルの予測等の精度等の定量的な指標によって評価可能なケースと異なり、テストケースに対してその生成物が妥当かどうかの評価は人間による定性評価の依存度が高くなる。そもそも品質をどの観点で評価すべきか、は個々のユースケースに依存するため、**4 調達実施前時のテキスト生成 AI 固有の留意点**で考察する。ここでは定性評価の実施全般のリスクである、評価者の主観に基づくバイアスの混入、費用や開発リードタイムの増加を軽減する方法を提案する。

軽減策

(1) 知識の有無を選択問題によって評価する

正しい知識が反映されているかのみを確認する場合は、選択問題を用意しテキスト生成 AI を使ったシステムに回答させるといった方法をとることで、基準を明確にできる。この水準まで明確化すると定量的な評価が可能になり、評価の自動化ができる。

イ 評価者の主観に基づくバイアスに起因するリスク

テキスト生成 AI の成果物を人間による定性評価を行う場合、その評価者の主観、つまり知識や経験や文化的背景の違いなど多様な要因により、なんらかのバイアスが発生する。その結果、期待された品質を達成していないことが実用化してから発覚するリスクが起これる。ここではそのリスクを軽減する方策を提案する。

軽減策

(1) 評価観点や基準を明確にする

テキスト生成 AI の生成物の良し悪しをただ漠然と評価させるのではなく、問題や課題点を分解して、評価観点や基準を明確にすることで、主観に基づくバ

バイアスを抑止できる。例えば、検索拡張生成（RAG, Retrieval Augmented Generative）の技法、つまり文章検索で関連文章を抽出し、それを大規模言語モデルにプロンプトとして渡す手法を用いる場合は

- (1) 質問文に対して適切な関連文章が抽出できているか
 - (ア) もし想定していた関連文章があれば、それをちゃんと抽出しているか（再現率の観点）
 - (イ) 抽出した関連文章に不適切なものが含まれていないか（適合性の観点）
- (2) 関連文章に対して大規模言語モデルの出力が適切か
 - (ア) 出力結果に含まれている固有名詞は関連文章内に含まれているか
 - (イ) 出力結果の内容は関連文章内で言及されているものか
 - (ウ) 複数の関連文章の結果を統合している場合、その論理の繋がり
は自然か
- (3) 最終的な生成物は求められる品質をクリアしているか
 - (ア) 十分読みやすいか（ユースケースに応じてもっと観点を明確にする）
 - (イ) 出力形式は意図通りになっているか

等の観点や評価項目が考えられる。分解したところで、主観に基づくバイアスの影響は完全に除外できないが、分解しない場合と比べて大幅に抑制可能である。

(2) 複数人により独立に実施する

1つの項目に定性テストを行う人間を複数人アサインし、それぞれが独立に評価した内容を総合することで主観に基づくバイアスを抑止できる。いわゆるダブルチェック、トリプルチェックである。

ただし、その複数人が同じ種類のバイアスをもっている時はリスクが軽減せず、人件費や開発リードタイムは明確に増加するといった欠点も多い。

ウ 費用や開発リードタイムの増加に起因するリスク

定性テストを実施するにあたって、人員をアサインした場合にその人件費が増加し、また定性テスト自体も時間のかかる工程であるため開発リードタイムの増加を招きやすい。これはプロジェクト管理の観点から明確なリスクである。

軽減策

(1) 機械的な品質評価手法を導入する

人手によらず機械的な品質評価を行うことで、費用や開発リードタイムを抑制することができる。テキスト生成の文脈では、機械的評価手法を大別すると以下の2種類がある。

(1) n-gram 等の深層学習の発展以前から用いられてきた技術をベースとするもの

(2) Transformer 等の深層学習をベースとするもの

この両方とも正解のテキストデータ、つまり入力に対して尤もらしい文章を事前に用意しておく必要がある。

まだまだ発展途上の分野であることから詳細は付録の **8 文章間の類似性の機械的評価方法について** で述べる。

(2) 人間の評価者が行うテストケースを絞り込む

単純に品質評価の対象のテストケースを絞り込めば、その分だけ品質評価に関する費用や開発リードタイムを削減することができる。当然ながらその分だけテストが行われなくなることになり、プロジェクトの品質要件を満たさないシステムがリリースされる危険性は高くなる。そもそもテストケース自体のカバレッジを十分に上げることが困難であり、その解説を **5.2.3) テストケースのカバレッジが不十分なことに起因するリスク** の節で行う

人間の評価者が行うテストを、機械的品質評価の結果を用いて特定の基準で絞り込むことで、一部だけを実施するといった、機械的品質評価と組み合わせた絞り込み方法を用いる手法も考えられる。この方法により、テスト対象のテストケースを減らすことなく、人間の評価者が行うケースを合理的に減らすことができる。特に、評価者に高度な専門知識を求めるケースでは有益な方法である。

3) テストケースのカバレッジが不十分なことに起因するリスク

テストケースごとの品質評価が仮に完璧に実施できたとしても、テストケース自体が十分でないことに起因し、プロジェクトの品質要件を満たさないシステムがリリースされるリスクが存在する。このリスク自体は情報システムの開発において一般的なものであるものの、テキスト生成AIでは入力が自然文に近いケースが大半であると考えられ、テストケースのカバレッジを十分確保することが困難であり、その軽減策も独特であるため本文書に含めた。

想定リスク

テキスト生成AIの入力、特にプロンプトには自然な文章が入力され、文中のごく一部が違っただけで生成物が大きく変化する可能性がある。この特性から

正常系のユースケースの入力を洗い出すことが不可能に近く、理想的なテストケースと比べてかなり狭い範囲でしかテストできない。

テキスト生成 AI 以外のシステムでは、キーワード検索システムがこれに類する。しかし、以下の 2 つの条件の違いにより、テキスト生成 AI を用いたシステムの方がテストケースのカバレッジの確保がより困難になっている。

- (1) 検索システムで想定される入力（クエリ）のキーワードの集合よりも文章の多様性が高い
- (2) クエリやアイテムのベクトル表現化を行わない古典的なキーワード検索システムではクエリにキーワードを一つ追加した時のシステムの挙動が予測できたが、テキスト生成 AI では挙動の予測が不可能に近い

軽減策

(1) 過去に実際にあったやりとりをそのままテストケースに含める

すでに存在するシステムや業務フローの一部をテキスト生成 AI に置き換える場合、過去に実際にあったやりとりをそのままテストケースにすることができ。例えばヘルプデスク業務の一部をテキスト生成 AI で置き換える場合は、過去に実際にあった問い合わせ内容とそれに対する回答を、プロンプトと正解データとしてテストケースに利用できる。

この手法では安価に大量にテストケースを入手できる。しかし、過去のやりとりをテストケース化しやすい形式で保存していない場合や、全く新規のシステムや業務フローを構築する際にはこの手法が使えない。

(2) 作成したテストケースの入力文の言い換え処理をテキスト系生成 AI に実施させる

テキスト生成 AI ではプロンプトが 1 文字異なっただけで出力する結果が大きく変わる可能性がある。この性質から、とあるプロンプトでそこからの生成物の品質評価を実施したとしても、同様の意味合いをもつプロンプトでの生成物の品質は保証できない。そこで、テキスト生成 AI を用いてテストケースの入力文を言い換えさせることでテストケースを増加させる手法を採用することがある。この手法を用いることで言い回しが異なったケースを評価できるようになる。

この手法のデメリットは、テキスト生成 AI 実行に関するコストが増えることである。また、評価時のエラーの原因に「テキスト系生成 AI の言い換えミス」が含まれるようになり、言い換え処理自体の品質が一定以上でないテストに関するコストがかえって増大する可能性もある。

(3) 事前にテストしたケースのみでテキスト生成 AI の生成物を利用する

テキスト生成 AI の生成物の品質を事前にテストできていない範囲がある場合は、その範囲において生成物を用いないことによりリスクを回避できる。この方法を採用すると、テキスト生成 AI の利用はすべてバッチ処理で行い、オンラインではテキスト生成 AI を使わないことになる。オンラインでテキスト生成 AI を使わない、といった選択をとることで、インフラコストやレスポンスタイムの改善等の副次的な効果も見込める。

この手法のデメリットは、テストケースが十分な量になるまで、実証時にテキスト生成 AI による恩恵がほとんど受けられないことである。

4) テスト済みの生成物のみを用いる場合の工夫

今までの議論の繰り返しになるが、テキスト生成 AI の品質保証を実現する最も安全な手法は、テストしていないケースを用いないこと (5.2.3-(3) 事前にテストしたケースのみテキスト生成 AI の生成物を利用する) である。ここでは、今までの議論の関係性を明確にするためのまとめを行う。

この手法を最大限利用するために大量のケースでテストすることが重要である。その手法には以下のものがある。

- (1) テスト対象のケースを増やす
 - (ア) 5.2.3) (1) 過去に実際にあったやりとりをそのままテストケースに含める
 - (イ) 5.2.3) (2) 作成したテストケースの入力文の言い換え処理をテキスト系生成 AI に実施させる
- (2) 大量のテストケースを評価する
 - (ア) 5.2.2) ウ (1) 機械的な品質評価手法を導入する
テストケースを増やすだけではなく、テストケースあたりの評価の品質向上も重要である。その手法には以下のものがある。
- (3) 生成物のばらつきに対処する
 - (ア) 5.2.1) ア (1) ばらつきに関するパラメータを調整する
 - (イ) 5.2.1) (3) 同じテストを繰り返し行う
- (4) 評価方法を工夫する
 - (ア) 5.2.2) ア (1) 知識の有無を選択問題によって評価する
 - (イ) 5.2.2) イ (1) 評価観点や基準を明確にする
- (5) 評価者のアサインを工夫する
 - (ア) 5.2.2) イ (2) 複数人により独立に実施する
 - (イ) 5.2.2) ウ (2) 人間の評価者が行うテストケースを絞り込む

その他にも、3.1.2) c) 過去のテキスト生成 AI の生成物を再利用す

る の手法を使うことで、サービス利用者からの入力文章がそのままの形でテストケースに含まれていなくても、類似した想定入力文章に対応するテスト済みのテキスト生成AIの結果を返す、といった工夫を行うことで、現実的な数のテストケースでも実用的なシステムを構築しやすくなる。

この入力文章とテストケースの判定手法は他にも、**5.2.2) ウ (1) 機械的な品質評価手法を導入する** で述べた方法や、入力文章に形態素解析、品詞フィルタリング、固有表現抽出を行い、その結果に基づくキーワード検索結果に基づいた回答結果をテストケースに適用する等の手法が考えられる。この手法を用いた場合、**5.2.3) (2) 作成したテストケースの入力文の言い換え処理をテキスト系生成AIに実施させる** の必要性は大きく下がる。

この考え方は、テキスト生成AI以前のチャットボットでの応答シナリオの充実にのみテキスト生成AIを使う、といった考え方と同じである。チャットボット以外でのユースケースでも、テキスト生成AIの利活用のスコープを適切に限定することは、品質の安定だけでなく、開発時や運用保守時のコストカットや、レスポンスタイム向上によるユーザビリティの改善にもつながるため、まずは「本当にテキスト生成AIが必須の処理なのか」「他の自然言語処理技術で代用できないか」の検討を行うことを推奨する。この検討の具体例は**3.1.2) (2) テキスト生成AI以外の手法を採用する** で簡単に紹介した。

5.3 5章 まとめ

設計・開発時には、テキスト生成AI固有のリスクを特に注視する必要がある。本章では、生成物の品質向上とその評価に重点を置いたリスク軽減策を提案した。特に、テキスト生成AIをバッチ処理でのみ用い、テスト済みの結果しか返さない形でのチャットによる情報検索のあり方の提案について述べた。適切な品質保証プロセスの確立が、信頼性の高いシステム構築に不可欠であり、継続的な品質管理が重要な役割を担う。本章がその確立の一助となれば幸いである。

6 サービス実施時に留意すべきテキスト生成AI固有の留意点

本章の目的は、サービス実施時におけるテキスト生成AIの利活用に伴う固有のリスクを管理し、業務の実績データや非機能要件の達成度を評価する手法を提供することである。スコープは、サービスが稼働し始めた後の実際の運用環境でのモニタリングと、継続的な改善サイクルの方法論に焦点を当てている。これはテキスト生成AIをどのように利活用するかによって観点が異なるため、ユースケースごとに述べる。

6.1 業務の実績データや利用者要望を分析できているか

1) チャットでの利用の場合

ここではユースケースのうち、チャットインターフェースで利用者サービス利用者とインタラクティブに対話する機能としてテキスト生成AIをオンライン処理で用いるケースについて述べる。一般的なシステム開発と同様に、予算要求時に設定したKPIを測定する。このユースケースで特に留意すべき点は、サービス利用者からのチャット入力文の実績データの取り扱いである。チャットインターフェースでテキスト生成AIとインタラクティブにチャットをすると、サービス利用者の体験としてはヒト相手にメッセージを送るものに近くなる傾向にある。そのため、利用規約やプライバシーポリシーで明記してあったとしても、テキスト生成AIへの入力文をシステム管理者が実績データとして確認すること自体をプライバシーの侵害とみなされるリスクがある。このリスクの軽減策として、実績データの取り扱いをプライバシーに配慮した形で定めそれを厳密に運用するのは大前提とし、サービス利用者の感情面を配慮したサービスデザインを熟慮することが挙げられる。

2) バッチ処理での利用の場合

ここではユースケースのうち、大量の文章に対してラベル付けやテキストデータ変換、その他翻訳や要約や文章作成等の自然言語処理を行う機能としてテキスト生成AIをバッチ処理で用いるケースについて述べる。一般的なシステム開発と同様に、予算要求時に設定したKPIを測定する。非機能要件として、最大同時リクエスト数やリクエスト成功率、目標処理時間等のシステム指標が求められているはずなので、それらを適切な基準で監視する。

3) 情報検索での利用の場合

ここではユースケースのうち、情報検索を目的としたサービスで検索エンジ

ンの補助としてテキスト生成 AI をオンライン処理で用いるケースについて述べる。一般的なシステム開発と同様に、予算要求時に設定した KPI を測定する。機能要件として、最大同時リクエスト数やリクエスト成功率、レスポンスタイムのサービスレベル等のシステム指標が求められているはずなので、それらを適切な基準で監視する。それ以外にもテキスト生成 AI を導入したことによる検索結果の品質改善への介入効果を適切に行うため、可能であれば A/B テストを積極的に行う。A/B テストでは、例えばテキスト生成 AI を導入した場合としなかった場合の 2 群を用意し、それぞれに対してランダムにサービス利用者を割り当て、2 群間でのクリック率や直帰率等の行動指標に発生した差を統計的に確認する。テキスト生成 AI を導入することにより運用コストが増大するので、その費用対効果が妥当なものであるかはしっかりと分析すべきである。

4) 自然言語からのプログラム作成利用の場合

ここではユースケースのうち、ダッシュボードやソースコード等をサービス利用者の自然言語により記述可能にする機能としてテキスト生成 AI をオンライン処理で用いるケースについて述べる。一般的なシステム開発と同様に、予算要求時に設定した KPI を測定する。特に注目すべき指標は(1)サービス利用者の自然言語から適切なアウトプットが出力された割合(2)テキスト生成 AI により業務時間がどこまで短縮できたか、である。

(1)サービス利用者の自然言語から適切なアウトプットが出力された割合に関しては、不適切な結果であった場合にフィードバックされる機構が欲しくなるが、SaaS 製品を導入している場合難しい場合も多い。そこで、アンケート調査や内部向けの報告窓口の設置によって間接的に推定する方法を検討する。

(2)テキスト生成 AI により業務時間がどこまで短縮できたかの測定は、実際に職員にテキスト生成 AI の有無の条件ごとに作業させて測定する手法が比較的正確だが、実験実施の負担が大きい。そのため現実的にはヒアリングやアンケート調査に頼らざるを得ない場合が多い。そもそもダッシュボードの操作やソースコード作成が元々できない職員の場合の、業務短縮効果は、そのスキルの平均取得時間を含められる。ただし、このスキル取得に係わる時間を業務短縮効果に含められるのは、その職員の最初の 1 回の作業だけであり、重複カウントしないようにする。

6.2 非機能要件に関する実績データを把握しているか

1) チャット利用の場合

ここではユースケースのうち、チャットインターフェースで利用者サービス利用者とインタラクティブに対話する機能としてテキスト生成AIをオンライン処理で用いるケースについて述べる。

サービス利用者からの実際の入力がログの形で入手可能になり、それらがテストケースに含まれていなかった場合、テストケースに追加することができる。これは4.1 [チャットでの利用の期待品質](#)でも述べた。この実際の入力をベースとしたテストケースを用い、改めて品質評価を行うことで非機能要件に関する実績データが得られる。ただし、6.1 [業務の実績データや利用者要望を分析できているか](#)で述べたように、重要な実績データであるサービス利用者からの入力文の扱いはプライバシー侵害だとみなされないように注意する。

2) バッチ処理での利用の場合

ここではユースケースのうち、大量の文章に対してラベル付けやテキストデータ変換、その他翻訳や要約や文章作成等の自然言語処理を行う機能としてテキスト生成AIをバッチ処理で用いるケースについて述べる。4.2 [バッチ処理での利用の期待品質](#)で述べたテストケースを実績データから増やし、それをテストすることで、非機能要件に関する実績データが得られる。

3) 情報検索での利用の場合

ここではユースケースのうち、情報検索を目的としたサービスで検索エンジンの補助としてテキスト生成AIをオンライン処理で用いるケースについて述べる。4.3 [情報検索での利用の期待品質](#)で述べたテストケースを実績データから増やし、それをテストすることで、非機能要件に関する実績データが得られる。

4) 自然言語からのプログラム作成での利用の場合

ここではユースケースのうち、ダッシュボードやソースコード等をサービス利用者の自然言語により記述可能にする機能としてテキスト生成AIをオンライン処理で用いるケースについて述べる。4.4 [自然言語からのプログラム作成での利用の期待品質](#)で述べたテストケースを実績データから増やし、それをテストすることで、非機能要件に関する実績データが得られる。

6.3 業務で扱うデータの品質を維持しつつ適切なライフサイクル管理を行っているか

一般的なシステム開発のデータ品質とライフサイクル管理はデータの不整合の発生抑止に重視が置かれるが、ここではAIプロジェクトの観点でのデータのライフサイクルについて述べる。これは、(1)実際の利用実績が当初想定から乖離あるいは変化、(2)利用した機械学習モデル(ここでは大規模言語モデル)の変化、の2つの変化により機能要件・非機能要件を満たさなくなるリスクのことを指す。機能要件が満たされているかの監視は一般的なシステム開発でも共通した話であり、非機能要件が満たされているかの監視については**6.2 非機能要件に関する実績データを把握しているか**で述べたので割愛する。

機能要件・非機能要件が満たされなくなる前に気が付く、満たされなくなった後に原因分析を行い適切な対策をとる、の両方のために、原因となる変化を観測することは非常に重要である。仮にシステムが機能要件・非機能要件を満たさなくなった場合、その要件を再び満たすための改修だけでなく、サービスの停止や終了も視野にいった検討も視野に入れるべきである。

(1)実際の利用実績が当初想定から乖離あるいは変化したか、についての観測は**6.1 業務の実績データや利用者要望を分析できているか**で述べた。特にサービス実施直後は、設計・開発時との想定との乖離がないかをしっかりと確認することを推奨する。

(2)利用した機械学習モデル(ここでは大規模言語モデル)の変化については、テキスト生成AIの機械学習モデルが直接提供されるケース以外の利用形態については注意すべき点であり、付録の**7.1 4) アップデート対応に関するリスク**と**7.2. 4) アップデート対応に関するリスク**を参照していただきたい。

6.4 6章 まとめ

サービス実施時には、実際の運用データをもとにした継続的な品質評価が必要である。本章では、業務実績データの分析と非機能要件の達成度の検証に焦点を当て、適切なライフサイクル管理を通じて、サービスの持続可能性を保証するための留意点を強調した。これらの実施により、テキスト生成AIの持続的な価値提供が実現できる。

まとめ

本文書ではテキスト生成AIの各種提供形態とユースケースに基づいたリスクを検討し、それに対する軽減策やその留意点を述べた。一方で、これらがより実践的な内容になるためには、実行可能なフレームワークやチェックリストの形式をとる方が望ましいが、本文書はその水準にまだ達していない。あくまで留意点を述べているだけである。特に下記の箇所では実践的なフレームワークやチェックリストの整備が重要であると考えている。

- テキスト生成AIの提供形態の選択プロセス
 - 利用シナリオ、セキュリティ要件、費用対効果等を考慮した提供形態決定の基準
- コスト削減戦略の具体化
 - 各提供形態におけるコスト削減の具体的な方法と手順を含んだ、コスト削減のためのアクションプラン等
- リスク管理計画
 - リスクの特定、評価、対応策の立案を含む詳細なプロセスやチェックリスト等
- 非機能要件のテストと評価
 - 非機能要件に関する評価基準やテスト方法等
- 品質保証プロセス
 - 品質を評価し、維持するための具体的な手法やツール等
- サービス利用者フィードバックの収集と分析
 - フィードバックの体系的な収集と分析、そしてそれをシステム改善にまとめつけるためのプロセス等
- 運用実施時の品質管理
 - 責任体制を明確にするための、運用実施時における品質管理のプロセス等

現段階（2024年5月現在）では、フレームワークやチェックリストを作成できる水準での知見の蓄積が不足している。今後、検証を勧めていく中で知見が更新され、実用的なフレームワークやチェックリストを追記する改定を目指している。

付録

特に、技術的に込み入った内容かつ変化が早い領域について付録の形で紹介する。

7 提供形態ごとの想定リスク

本章の目的は提供形態による違いに伴うリスクの想定とその軽減策を示すことである。スコープは、ライセンス、調達容易性、機密性、アップデート対応、ベンダーロックイン、コストマネジメントの 6 つの観点に焦点をあて、想定リスクと軽減策を整理する。

7.1 テキスト生成 AI を組み込んだサービスとして提供されるケースに関するリスクとその軽減策

テキスト生成 AI が何らかのサービスに組み込まれた状態で提供される利用形態を想定する。ChatGPT や Github Copilot のような SaaS 型のサービス以外にも、Bing Chat や Google Bard といった広告課金型のサービスも該当する。テキスト生成 AI は有用な用途が広く、今後さまざまなサービスに組み込まれていくと考えられる。

2024 年 4 月現在、各府省庁で生成 AI を利用したサービスで要機密情報を扱う条件に (1) 必要なセキュリティ要件が守られる契約を実施 (一般的に約款型では厳しい) (2) クラウドサービス (参考: [政府機関等の対策基準策定のためのガイドライン \(令和 5 年度版\)](#) の 4.2 クラウドサービス) に関わる係る関連規程に基づく対応 (政府情報システムのためのセキュリティ評価制度 (ISMAPP) による認証が一例) を行ったうえで、AI 戦チームに報告・了承をとる必要がある。(参考: [ChatGPT 等の生成 AI の業務利用に関する申合せ \(第 2 版\)](#))

1) ライセンスに関するリスク

生成 AI 固有の規約として、生成物を競合他社による AI 学習利用を禁止している例が挙げられる。(参考: [OpenAI 社の利用規約](#))。

2) 調達容易性に関するリスク

広く市場に流通しているサービスの調達時に特別なリスクは考えにくい。ただし、サービスの背後で大規模言語モデルを Web API 経由で利用していた場合、Web API 側のアップデートにより以前同じものを調達できなくなるリスクが存

在する。詳細は [7. 2. 4\) アップデート対応に関するリスク](#) を参照いただきたい。

要機密情報を扱う場合、該当サービスが政府機関等のサイバーセキュリティ対策のための統一基準群等のセキュリティ要件を満たすかには注意が必要である。

3) 機密性に関するリスク

機密性の議論は、個人情報を含めた要機密情報を扱う場合のセキュリティ要件の一つでしかなく、テキスト生成AI固有のリスクではない。テキスト生成AIでは、教師データとして与えた文章らしい出力を返すような機械学習が行われるため、ユーザからの入力をAI開発者が教師データとして扱うことによる機密流出が問題視されてきた。しかし、テキスト生成AIへの入力の機密性が守られないケースは機械学習であること以外にも多岐にわたり、利用時に意識することもサービス契約者との契約面やシステムがセキュアであるかが主であるため、あまり「学習」という側面にとらわれすぎることなく、機密性が守られるかについての規約や契約の精査と、情報システムとして十分なセキュリティ対策が施されている必要があるかを確認する必要がある。[政府機関等の対策基準策定のためのガイドライン（令和5年度版）](#)の4.2 クラウドサービスを参考にしていただきたい。

4) アップデート対応に関するリスク

テキスト生成AIを活用したサービスは、その内部でOpenAI社、Anthropic社やAlphabet社等のAIモデルを、Azure、AWSやGCPといったクラウドサービスプロバイダのPaaSを利用していることが多い。このPaaS上のモデルがアップデートされた場合に、提供されたサービスに挙動が変わることがある。(参考：[OpenAI社のGPT-3.5とGPT-4の性能変化についてまとめている調査結果](#))

この利用形態では、アップデートに伴う挙動の変化を主体的に軽減することは難しいため、挙動の変化がリスクになるような使い方をしない、といった利用方針でカバーする方法を推奨する。[5. 2 テキスト生成AIによる生成物の品質評価・品質保証についてのリスクとその軽減策](#) も関連した内容を含むため参考にしていただきたい。

5) ベンダーロックインに関するリスク

テキスト生成AIを用いたサービス固有の特徴として、ロジックの大部分が特定の大規模言語モデルに依存しているサービスが広く出回っている。特にChatGPTに代表される、チャットインターフェース経由でユーザが大規模言語

モデルを利用することを目的にしたサービスが挙げられる。この種のサービス利用時にベンダーロックインが発生してしまうと(1)他のより優れた大規模言語モデルが利用できない(2)より安価な類似サービスに乗り換えることができず unnecessary コストが発生する、等のリスクが考えられる。

軽減策

チャットインターフェース経由でユーザが大規模言語モデルを利用することを目的にしたサービスを利用する際に、ベンダーロックインを防ぐために

- (1)利用している大規模言語モデルはバージョン情報を含めて明示化されているか
- (2)大規模言語モデルに入力されるプロンプトの一部やパラメータが隠蔽されていないか

の確認が重要である。これらの情報が開示されていれば、他のサービスでもテキスト生成 AI に対して同じような挙動をさせることができる。

また、過去のチャット履歴の保存機能や、プロンプトをテンプレートとして登録する機能がある場合等(参考: [ChatGPT では GPTs が該当する](#))は、それらのデータをエクスポートする機能等のデータの囲い込みを軽減する機能もあると、よりベンダーロックインを回避しやすい。

大規模言語モデルごとの最適なプロンプトの違い等の議論は、**7. 2 5) ベンダーロックインに関するリスク** も参照いただきたい。

6) コストマネジメントに関するリスク

SaaS 型の製品を利用する場合、大半の場合でコストがアカウント数とプランに比例し、コスト予測は容易である。一方で費用対効果が見合うかの試算が非常に難しい。SaaS 型の製品を利用し、行政職員の業務改善効果を期待する場合、その効果測定が重要となる。これはテキスト生成 AI 固有の話題ではなく DX や BPR といった分野一般の話題であるため本書では割愛する。

7. 2 テキスト生成 AI を利用するための Web API がクラウドサービスとして提供されるケースに関するリスクとその軽減策

ここでは、クラウドサービスとしてテキスト生成 AI の大規模言語モデルが提供される利用形態を想定している。具体的には、OpenAI 社の GPT シリーズ、Microsoft 社の Azure OpenAI Service、Amazon 社の AWS Bedrock や Alphabet 社の GCP Vertex AI 等を想定している。これらのサービスでは Web API として大規模言語モデルを主に従量課金、一部で定額課金、で利用することができるだ

けでなく、Web API を便利に使うための Web アプリケーションもセットで提供されることが多い。

1) ライセンスに関するリスク

テキスト生成 AI 固有でなく、サービス利用時の契約に関する一般的なリスクであるため割愛する。生成 AI 固有の規約としては、生成物の競合他社による AI 学習への利用を禁止している点が挙げられる。(参考: [OpenAI 社の利用規約](#))

2) 調達容易性に関するリスク

一般的に市場に流通しているサービスを念頭においており、調達は容易のためリスクは見当たらない。ただし、生成 AI に関連するクラウドサービスの最先端の機能ではアクセスが制限されていたり、利用可能なリージョンが限定されていたりと制約が多く、その制約のために調達が容易でない場合もありうる。また、大量の計算リソースを必要とする場合は、クラウドサービスの利用であってもその計算リソースを確保できない状況もありうる。

3) 機密性に関するリスク

テキスト生成 AI 固有でない AI システムの一般的なリスクのため割愛する。一般的な AI システムと同じく学習用のデータや推論用のデータがどこにどのように保存されるかとそれに伴う想定リスクについての整理が重要である。

4) アップデート対応に関するリスク

従来の AI システムと比べて、事前学習済みの大規模言語モデルを追加学習無しでも十分に利用できるケースが多い。その事前学習済みの大規模言語モデルの利用を前提としていると、その日々アップデートや保守終了 (End Of service Level、EOL) によって、自分達で十分にテストできていない大規模言語モデルを利用せざるを得ないケースが起こりうる。特に事業者や国民向けの政府情報システムではリスクが大きいと予想される。

軽減策

(1) テスト済みの結果しか返さない

もっともリスクが軽減される方法である。5.2 テキスト生成 AI による生成物の品質評価・品質保証についてのリスクとその軽減策、特に 5.2.4) テスト済みの生成物のみを用いる場合の工夫 を参考にいただきたい。

リスク影響の評価

大規模言語モデルが EOL を迎える前に、移行候補の大規模言語モデルで十分な性能かどうかを検証できていれば、アップデートに関するリスクの影響を事前に評価できる。リスク影響が十分に低ければ特に軽減策を取る必要がなくなる。一般的な傾向として大規模言語モデルはより良い性能になるよう改良されていくものなので、軽減策にコストをかける前にリスク影響を見積もることを推奨する。

(1)新しい大規模言語モデルを開発内部でテストする

新しい大規模言語モデルに実際のプロンプト等の入力を入れて、その生成物の品質が許容できるものかを評価する。[5.1 テキスト生成 AI の生成物の品質面への懸念](#) を参考にしていきたい。

(2)ユーザビリティの悪化を検知する

開発者による内部テストではなかなか評価しにくい総合的なユーザビリティの評価は、実際に新しい大規模言語モデルを段階的にリリースすることによる A/B テストが有効である。A/B テストとは、サービス利用者をランダムに違う条件に割り振り、その条件毎で利活用のされ方を定量的に比較する手法である。A/B テスト自体はテキスト生成 AI 固有の評価方法ではないので詳細は割愛する。

5) ベンダーロックインに関するリスク

他のクラウドサービスプロバイダの Web API に切り替える場合、以下の 3 点の対応が必要となる。

- (1) 異なる大規模言語モデルを利用する場合、プロンプトのチューニング
- (2) Web API に送るリクエストボディのフォーマット
- (3) その他、エンドポイントやネットワークの設定、認証情報等

この切り替え対応の実施コストが極めて高い場合、ベンダーロックインに関するリスクが増大する。

(1)は大規模言語モデルによってプロンプトの効率的な書き方が異なる点に注意が必要である。(参考: [Anthropic 社の Claude では \n\nHuman: に特別に意味をもたせている](#))

(2)はテキスト生成 AI 固有の内容でないため割愛する。パラメータ名やとりうる値が異なる等の違いはあってもクラウドサービスプロバイダ間で似ている傾向がある。

(3)はテキスト生成 AI 固有の内容でないため割愛する。

実装レベルでクラウドサービスプロバイダの Web API の違いを意識しないで済む方法の一つに LangChain 等の開発フレームワークの導入も考えられる。しかしフレームワークを導入してしまうと、今度はそのフレームワークにロックインされるリスクが生じる。まだ発展途上かつ発展が早い分野であるため、情報システムの開発体制やエンジニアのスキル等も考慮しつつ、フレームワークの安易な導入といった早すぎる最適化をしないよう慎重に検討することを推奨する。

軽減策

テキスト生成 AI 固有でない部分、例えばソフトウェア設計時にテキスト生成 AI サービスを利用するコンポーネントを適切に分離し、そこだけ変更した場合の評価実験を行いやすくする等のマイクロサービス一般の工夫については割愛する。

テキスト生成 AI 固有の部分に関しては、基本的に十分にテストができれば問題ないため、**7.1.4) アップデート対応に関するリスク** と同じ内容である。新しいクラウドサービスプロバイダのテキスト生成 AI サービスを利用する場合のリスク全般は本章の他の節で記載されている。

7.2.6 コストマネジメントに関するリスク

大規模言語モデルをクラウドサービスプロバイダの Web API で利用する場合、その料金形態が独特である点に注意が必要である。これは大規模言語モデルの推論時に、入力のプロンプトや出力される生成物の文書量のサイズが増えるとともに、必要とされる計算リソースが増える傾向がある性質に由来していると考えられる。入力のプロンプトよりも生成物の文書の方がサイズによる計算リソースの増加影響が大きい。そのため、多くの場合、入力分のサイズ（一般的にはトークン数、一部文字列長のケースもある）と出力文のサイズのそれぞれに比例した課金形態となっている。この料金形態によりリクエスト数だけでなく、リクエストの中身やその結果の出力にも大きく費用が依存する形になり、コスト予測が難しくなっている。そのため見積もりミスによる予算超過や利用停止のリスクが起こりうる。

軽減策

(1) 見積もりの精度を上げる

事前テストの結果等から入出力のテキストトークン数や文字数の悲観予測をしておく。トークン数の計算方法は各クラウドプロバイダーから提供されている場合も多く事前に確認しておく。

(2) 事前にテスト済みのものしか返さない

テキスト生成 AI による出力結果は事前にテストしたものしか返さない方式を採用することでテキスト生成 AI 固有のコストマネジメント上のリスクを大幅に軽減することができる。5. 2 テキスト生成 AI による生成物の品質評価・品質保証についてのリスクとその軽減策、特に5. 2 4) テスト済みの生成物のみを用いる場合の工夫 も参考にしていきたい。

(3) 予算超過を防ぐ

テキスト生成 AI の Web API 方式の提供では、出力トークン数(文字数)の上限をパラメータで設定できることもできる。このパラメータと入力文字数の制限を設けるプログラムを手前に挟むことで 1 リクエストあたりのコスト上限をコントロールできる。

リクエスト量の制限や、コストアラートの設定等の方法はテキスト生成 AI 固有でないクラウドサービス利用一般の話題のため割愛する。可用性をどこまで犠牲にできるかのトレードオフの関係になる点に注いただきたい。

クラウドサービスプロバイダによっては、テキスト生成 AI を従量課金ではなく定額で利用するプランを用意している場合もある。一般にこの契約は高価なため、テキスト生成 AI を大量に利用するケースで選択肢に上がる。

7. 3 テキスト生成 AI の機械学習モデルが直接提供されるケースに関するリスクとその軽減策

テキスト生成 AI を、特定のサービスに組み込まれた形ではなく、Web API の形式でもなく、大規模言語モデルを自身の環境にデプロイすることで利用するケースを考える。

このケースではインターネットに繋がらないオンプレの環境だけでテキスト生成 AI 利活用を完結することができるため、非常に機微な情報を利用する際等に選択肢に上がるが、初期構築や保守運用のコストが高く、利用できる大規模言語モデルも限定される、といったデメリットも存在する

1) ライセンスに関するリスク

ソフトウェアの利用のさいに、ライセンスを意識する必要があること自体はテキスト生成 AI 固有ではない。オープンソースとされる大規模言語モデルは、元となる大規模言語モデル (Meta 社の Llama2 等) を追加学習したものである場合があり、その場合は元となる大規模言語モデルのライセンスも継承していることが一般的であるため、注意が必要である。

生成 AI 固有の規約としては、生成物の競合他社による AI 学習への利用を禁止している点が挙げられる。(参考: [OpenAI 社の利用規約](#))

2) 調達容易性に関するリスク

生成 AI の世界的な盛り上がりから計算資源の確保、特に GPU、が難しくなっている。テキスト生成 AI を学習せずに利用するだけでも計算資源をかなり必要とするため、調達のコストが急騰したりそもそも調達不可能であったりするリスクが存在する。ただし大規模言語モデルの小型化の研究も盛んであるため、近い将来は必要とする計算資源が少ない大規模言語モデルでも品質要件を十分満たせるようになる可能性が高い。

3) 機密性に関するリスク

特になし

4) アップデート対応に関するリスク

特になし

5) ベンダーロックインに関するリスク

利用した大規模言語モデルが、他のクラウドベンダーでの利用を想定していない場合に強いベンダーロックインがかかるリスクがある。**7.2 4) アップデート対応に関するリスク**のように、その大規模言語モデルでなければいけないのかを正確に評価することが重要になる。

6) コストマネジメントに関するリスク

7.3 2) 調達容易性に関するリスク に述べたように非常に高価になるリスクがある。

8 従来型の情報検索サービスをテキスト生成AIにより改善する手法

ここでは、データベースや全文検索エンジンが既に存在する場合、それをテキスト生成AIにより改善する手法の一例の概要をまとめる。まだ発展途上の分野であるため頻繁な改訂を想定している。



図5. 情報検索システムでユーザビリティ向上を生成AIで実現できる箇所(再掲)

No	テキスト生成AI適用箇所	処理名称	処理内容	備考
1	回答候補の登録	回答候補自体と対応させる質問文の作成	引用させたいドキュメントを用いて、事前にQAを大量に生成する。	5.2.4) テキスト済みの生成物のみを用いる場合の工夫も参照いただきたい
2	回答候補の登録	ラベリング	サービス利用者が検索時にラベルで絞り込み検索ができるように回答候補にラベルをふっておく。このラベルはランキングアルゴリズムも特徴量に使うこともできる	ラベル自体の見直し頻度が少ない場合、テキスト生成AIでなく特化した教師付き学習モデルを開発することも検討する

3	回答候補の登録	可読性の高いテキストに変換	不要な HTML や改行コードの変換、検索結果画面に表示させるための要約テキストの作成等を行う	<p>不要な HTML や改行コードの変換はテキスト生成 AI なしでもルールベースでかなり実現できる</p> <p>テキストの要約はテキスト生成 AI 以外の手法もあるが、どのような要約にしたいかのコンテキストを含めたテキスト生成 AI 利用の方が良い場合も多い</p>
4	回答候補の登録	非構造データのテキスト変換	図表やイラストや写真等の情報もテキスト検索の対象にするためテキストでの説明を入れる	<p>マルチモーダルな大規模言語モデルの利用することで実現できる</p> <p>アクセシビリティ向上のための代替テキスト作成にもつながる</p>
5	回答候補の登録	同意語・表記ゆれ・シノニム等対策	回答候補のインデックスさせる文章（キーワード検索にひっかける用のもの）に元の文章の同意語・表記ゆれ・シノニム等を登録させる	<p>全文検索エンジンであれば辞書と設定ファイルを登録すれば、回答候補登録時にインデックスに反映され、クエリにも同じ処理が実施される</p> <p>辞書ベースのアプローチが望ましく、その辞書の作成にテキスト生成 AI を用いる</p>

6	サービス利用者の入力	固有表現抽出	サービス利用者の入力から固有名詞や重要だと思われる単語を抜き出し、その単語が正確に含まれる回答候補を検索上位に表示させる	テキスト生成 AI 利用でなく、形態素解析と辞書による方法を最優先に検討すること 辞書の作成にテキスト生成 AI を活用する方法が考えられる
7	サービス利用者の入力	同意語・表記ゆれ・シノニム等対策	サービス利用者の入力文に対して、同意語・表記ゆれ・シノニム等の処理を行う	No5 と同等
8	サービス利用者の入力	similarity search	サービス利用者の入力文章の意味を表現したベクトル表現 (embedding) に変換し、回答候補のベクトル表現との類似度を計算し、一定以上の類似度のあるものを検索結果に表示させる、または類似度が高い回答候補を検索上位に表示させる	9 文章間の類似性の機械的評価方法について で関連する要素技術に簡単に触れる ベクトル表現化のための機械学習モデルは厳密には生成 AI ではないが関連技術であるため、ここで紹介した 回答候補の登録時にもベクトル表現にしておく ベクトル表現に利用する機械学習モデルは基盤モデルをそのまま利用する以外にも fine tuning し用いる方法もある
9	サービス利用者の入力	カテゴリ推定	サービス利用者の入力文に対して事前に定	カテゴリ情報の利用は多岐にわたる

			<p>義したカテゴリを推定し、その推定情報を検索の処理に活用する</p>	<p>例えば、特定のラベルのついた回答候補を優先表示したり、No6, 8, 10, 11, 13のような処理をカテゴリ毎に返えたりが考えられる</p> <p>カテゴリ推定の手法は、サービス実施時にはテキスト生成AIではなく、従来の自然言語処理手法の採用の検討を推奨する</p>
10	サービス利用者の入力	意図解釈	<p>サービス利用者の入力から意図を読み取り、より適切な検索結果を提供するために入力を修正する</p>	<p>どのような入力か適切かを事前に定義する必要と、その適切な処理への写像の2つの処理が必要である</p> <p>その処理の構成要素の一つにテキスト生成AIの利用が選択肢になる</p> <p>変換後の入力文をサービス利用者に明示的に選択させるか否かで検索体験が大きく変わり、どちらが望ましいかはサービスの特性に依る</p>
11	システムの出力	回答結果の要約や解説	<p>検索結果の回答候補から、よりサービス利用者の入力に対して直</p>	<p>テキスト生成AIの知識ではなく、検索</p>

			接の返答になるような文章を生成する	結果の回答候補から文章を生成させる サービス利用者が元の文章を確認できるように引用元の情報の明記が望ましい 江戸川区のホームページ で導入された機能
12	システムの実出力	related query suggestion	サービス利用者の入力から、より適切な検索結果を提供するために入力文を提案する	内部処理自体はNo10と同様

表4 既存の情報検索システムをテキスト生成 AI で改善する一例

9 文章間の類似性の機械的評価方法について

文章間の類似性を自動的・定量的に測ることができれば、

- (1) テキスト生成 AI の生成物の品質評価の際に正解データとの比較
- (2) テキスト生成 AI の生成物の再利用時の際に利用すべき生成物の特定を大量かつ高速に実現することができる。

テキスト生成 AI の生成物の品質評価では、**5.2.2) ウ 費用や開発リードタイムの増加に起因するリスク** で述べたように、大量のテストケースで評価することが重要であるため、機械的な手法を導入しないと、試行錯誤の実施が困難になる。

テキスト生成 AI の生成物の再利用時の際に、過去の質問応答データの質問文章と入力文が十分に類似していると判定出来る場合は、対応する過去の応答データを出力する手法が考えられる。この手法によりテキスト生成 AI をオンライン処理で利用しないで済むことにより、品質の安定化や運用費用削減、レスポンスタイムの向上等の利点がある。**5.2.4) テスト済みの生成物のみを用いる場合の工夫** も参照いただきたい。

文章間の類似性を自動的・定量的に測るための機械的評価手法を大別すると以下の2種類がある。

- (1) n-gram 等の従来から用いられてきた技術をベースとするもの
- (2) Transformer 等の最新のニューラルネットワークをベースとするもの

この両方とも正解のテキストデータ、つまり入力に対して尤もらしい文章を事前に用意しておく必要がある。

n-gram 等の従来から用いられてきた技術をベースとするものは、例えば Bilingual Evaluation Understudy (BLEU) や Recall-Oriented Understudy for Gisting Evaluation (ROUGE) といった手法がある。これらは計算コストが少ないため評価の実施が比較的容易ではあるが、意味がほとんど変わらない語順変化や、否定語の有無、意味がほぼ同じ単語の言い換え等を苦手とし、総じて文の意図を理解した形での評価が困難であると言われることが多い。

一方で、Transformer 等の最新のニューラルネットワークをベースとしたものは何らかの形で文の意図を表現できているとみなすことができる挙動をするため、より洗練されたテキスト品質の評価手法であると考えられる。しかし計算コストが比較的高いため、常に Transformer 等のニューラルネットワークをベースとした評価手法を採用することが正解であるとは言い切れない。また、特定の固有名詞の有無を特に強調したい場合等はルールベースの手法と組み合わせることも考えられる。この Transformer 等のニューラルネットワークをベースとした評価手法も以下の2種類に大別される。

- (1) テキスト生成 AI の生成物と正解データのテキストをそれぞれベクトル表現化し、文章間の類似性を評価する
- (2) テキスト生成 AI の生成物と正解データのテキストをプロンプトに入れ、gpt-4 等の性能が良いとされる大規模言語モデルによって評価させる

ベクトル表現化には事前学習済みの Bidirectional Encoder Representations from Transformers (BERT) 等が使われ、最も単純な比較方法にはベクトル表現間のコサイン類似度が使われる。この手法自体も (1) 文章をどのようにベクトル表現化するか (2) ベクトル表現化するための機械学習モデルをどのように作るか (3) ベクトル表現化した後にどのように比較し指標化するか、等で多くの工夫の余地があり、Bilingual Evaluation Understudy with Representations from Transformers (BLEURT) といったより洗練された手法も提案されている。ルールベースの手法を採用するのではなくベクトル表現化するための機械学習モデルを fine tuning する選択肢も存在する。

テキスト生成 AI の生成物と正解データのテキストをプロンプトに入れ、テキスト系生成 AI によって評価させる手法は、特に追加学習を必要としない、いわゆる zero-shot Learning でもかなりの性能が出るとされている。この考え方で、MT-Bench や AlpacaEval、日本語だと Rakuda Benchmark 等のベンチマーク手法が開発されている (LLM-as-a-Judge と呼ばれる)。一方で、実施するための計算コストが非常に高いため、この方法を使った場合は費用や開発リードタイムの抑制効果が一番低い。

総じて、人間と機械的評価手法の間だけでなく、機械的評価手法の中でも評価の品質と実施のためのコストにトレードオフの関係がある。どの手法を採用するかはプロジェクトの目標に応じて異なるため一概に正解は決められない。一般論として、エンジニアがリリース前に手元で小さく改善する場合はコストのあまりかからない手法でテストと改善を高速に実施した方が良いが、求められる生成物の要件によっては簡易な評価手法は適さないケースも十分ありうるため、考慮を必要とする。