自己問答（Self-Check）による生成 AI のハルシネーション抑制の検証

—— 医薬品処方の禁忌チェックタスクを対象として——

永山 和樹　林 孝太朗　大和田 昭邦

株式会社 DynaxT 新技術商品開発部　株式会社 DynaxT　株式会社 DynaxT

2026 年 7 月 21 日

概要

病院では日々数百〜千件の処方箋が発行され、薬剤師がその内容をチェックしている。この業務は薬剤師数人がかりで行う負荷の高い作業である。これを AI 技術によって支援し、**薬剤師 1 人で処理できるようにすること**が本研究の大目的である。生成 AI は、RAG（Retrieval-Augmented Generation）を使用して資料を与え、「資料に基づいて回答せよ」と指示しても、誤解や創作によりハルシネーション (AI の嘘) を起こしうる。しかし医薬品処方の禁忌チェックにおいて、ハルシネーションによる誤判定は重大な医療事故に直結するため許容できない。本研究では、ハルシネーション抑制手法として**自己問答（Self-Check）**を採用し、医薬品医療機器総合機構（PMDA）の添付文書データ（約 1.8 万件）に基づいて処方の禁忌を判定できるかを検証した。100 種類の模擬処方に対する判定の結果、**事実に基づかない回答（ハルシネーション）は一件も発生しなかった。**また、想定解との一致率は **93%** であった。想定解とずれた 7% のうち、1% は法令知識の不足（本タスクのスコープ外)、残る 6% は「資料に基づけば問題ないが安全側に倒して疑義照会とした」ものであり、医学的・薬学的に危険な誤判定（見逃し）は存在しなかった。本稿では、実験環境・会話フロー・プロンプト設計・定量結果・考察を報告する。

目次

1　経緯

医薬品処方の禁忌チェックは、患者の安全に直結する薬剤師の中核業務である。判定の根拠となる資料としては、PMDA が公開する添付文書情報 [1] が整備されており、これを参照して判定を自動化できれば、業務負荷の軽減が期待できる。一方で、判定に AI を用いる場合にはハルシネーションが課題となる。RAG を活用したシステムであってもハルシネーションのリスクを完全に取り除くことはできない [3] とされており、禁忌チェックにおける誤判定は許容されない。そこで本研究では、その対策として、**自身の出力を自ら検証する**という自己問答（Self-Check）[4] の適用を検討する。

なお本検討は、**用途を特定した上での対策**であり、一般的な対話におけるハルシネーション対策を目的とするものではない。

2　環境

実験に用いたハードウェア・ソフトウェア構成を表 1 に示す。推論エンジンには vLLM[6]、根拠資料には PMDA 添付文書 [1]（ver. 20260215）を用いた。

表 1　実験環境

区分	項目	内容
AI サーバ (HW)	CPU	Xeon Gold 6326 ×2（32C/64T）
	RAM	1024 GB
	GPU	NVIDIA A6000（VRAM 48 GB）×10 枚（うち 8 枚割当）
AI サーバ (SW)	AI モデル	DynaAI-large-v1
	推論エンジン	vLLM v0.23.0
クライアント (HW)	CPU	Core i7-12650H
	メモリ	64 GB DDR4
	SSD	NVMe 1 TB
クライアント (SW)	Python	3.14.5
	DuckDB	1.5.4
	opencode	1.17.20
根拠資料	PMDA 添付文書	ver. 20260215

3　実験・検証方法

3.1　会話フロー

図 1 のような会話フローを実装した。ユーザから必要情報を取得する。AI は自己チェックを行いながら回答の生成を行う。自己チェックで問題が検出されなければそれを返答とし、検出された場合は以前の AI の回答を見ながら再推論してその結果を返答とする。得られた判定は想定解と比較する。

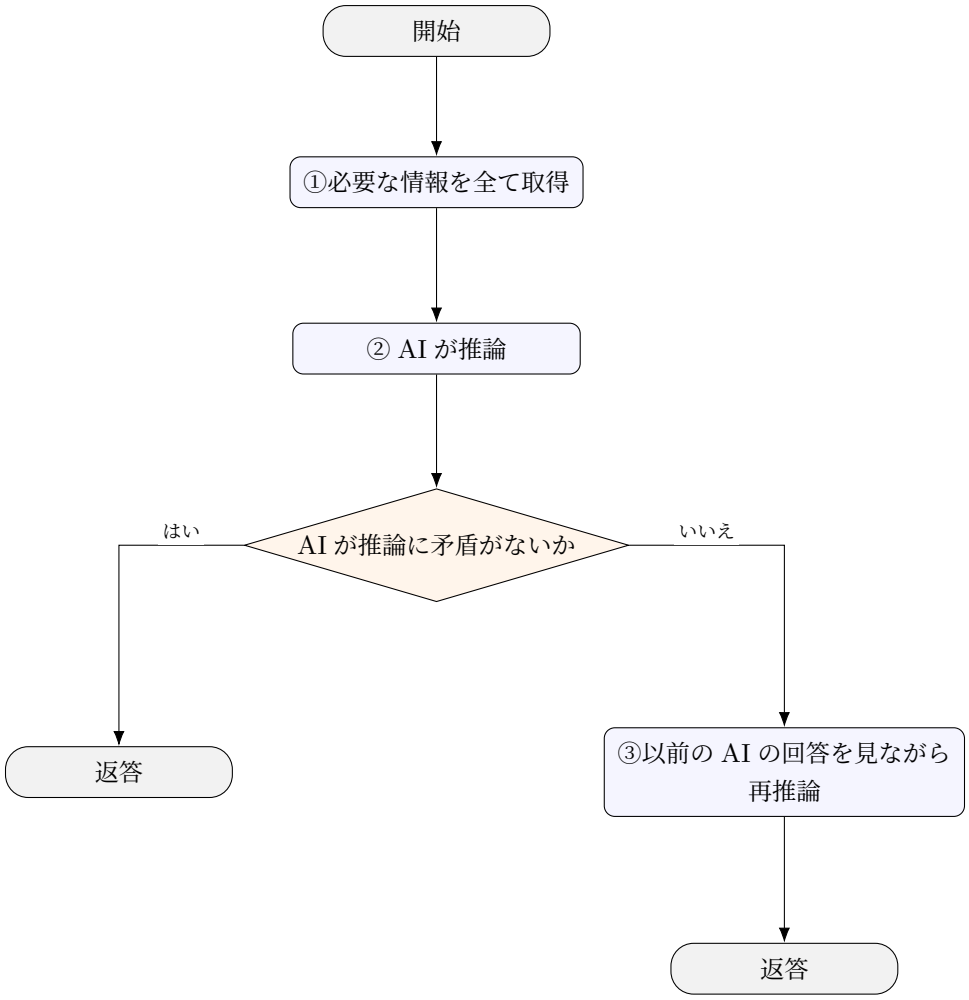

図 1　実装した会話フロー（図 3.1）。AI の推論に矛盾がないかを確認する。矛盾がある場合は再度推論して返答とする。

3.2　必要情報

判定に必要な情報は以下のように事前に定義している。

- 対象薬剤名
- 患者の年齢
- 現在服用中の全併用薬

3.3　プロンプト設計（要点）

AI に渡すプロンプトは、ハルシネーション抑制のため以下のルールを厳守させる。

1. 事実に基づかない情報を出力しない。断定できない場合は必ずその旨を明記する [5]。
2. 出典は必ず添付文書から確認する。根拠が見つからない場合、出典名を創作してはならない。
3. 入力に存在しない薬剤名・患者情報を勝手に補完・創作しない。
4. 実在が確認できない薬剤名の場合は禁忌を判定せず、その旨を明記する。

5. 疑わしい場合は**安全側**に倒せ。

エージェントには**約 1.8 万件の PMDA データの全文検索と、元ファイルへのパスを取得できるツール**（`kinki_search.py`）を与えた。総合判定 verdict は「禁忌／警告／要管理／併用注意／問題なし」から 1 つを選ぶ。

3.4 評価データ

計 100 種類の質問と想定回答を準備し生成 AI が出力した回答と比較する。

4 結果

100 問に対する判定を想定解と照合した。混同行列を表 2 に、予測精度の内訳を表 3 に示す。全体一致率は **93%** であり、**PMDA に明示的に記載されている禁忌を見逃すことはなかった。**唯一想定解を見逃した 1 件も、PMDA に関する内容ではなく法律関連の知識不足によるものであった。

表 2　混同行列（件数、$N = 100$）

		想定解（正解）		
		問題なし	**疑義照会**	計
AI 予測	**問題なし**	19	1	20
	疑義照会	6	74	80
	計	25	75	100

表 3　予測結果の内訳（表 4.1）

区分	割合	説明
正解	93%	想定解と一致
見逃し	1%	疑義照会の処方を「問題なし」と判定
厳しすぎ	6%	問題無い処方を「疑義照会」と判定

4.1 根拠取得の結果

判定の根拠（添付文書の該当記載）を取得できたかの内訳を表 4 に示す。

表 4　根拠取得結果（表 4.2）

区分	割合	内訳
根拠が取得できた	94%	うち 見逃し 0・厳しすぎ 6
一部の根拠が取得できなかった	4%	うち 見逃し 1・厳しすぎ 0
全く根拠が取得できなかった	2%	うち 見逃し 0・厳しすぎ 0

AI が根拠を取得できなかった場合を分析すると、キーワード検索（有効成分・薬名・用量など）で

根拠となる文書がヒットしなかったパターンである。複数回、手法を変えて根拠を取得しているため、全く根拠が取得できなかったケースは少なかった。全く根拠が取得できなかった場合は、常識に基づいて判断していた。

また、1 例のみ成分名が資料から全くヒットせず、代用として商品名で検索して使用したものがある。これは処方内容が成分名であるのに、同じものかを確約できないため要確認となっていた。商品名がインプットとして全く与えられていなくても、AI の持つ常識の範囲内であれば、成分名と商品名の間の変換ができる可能性がある。

4.2 想定解とずれた回答の分析

想定解とずれた 7 件を分析したところ、1 件（法令知識の不足）を除くすべてが、「**念のため確認しておいた方が良い**」というコメント付きでの疑義照会であった。これは、AI への指示の中で「少しでも疑惑がある場合は安全側にせよ」という指示があるためだと考えられる。

5 考察

根拠を取得する精度には、システム的な改善点が得られた。そのため、根拠を取得できる確率を向上させることができる。

一部、根拠が取得できていない状態でも判断はできるが、見落とす可能性がある。そのため、根拠が取得できなかった場合は、**何が取得できなかったかを明示した上で人間に判断を委ねる**ように変更する。

93% のケースでは判断根拠済みで思考過程を確認できるため、労力の削減に大きな効果があると考えている。生成 AI では、不審な点・根拠の薄い点について日本語でコメントを書くことができるため、そのコメントも人間の判断材料になることが期待できる。

■結論　自己問答（Self-Check）と、創作を禁じ安全側に倒すプロンプト設計、および PMDA 添付文書の全文検索ツールの併用により、本タスクにおいて**ハルシネーションによる誤判定は発生せず**、想定解一致率 93%・危険な見逃し 0 件を達成した。残課題は根拠取得の頑健性（表記揺れの正規化）と、根拠未取得時の人間へのエスカレーションである。

参考文献

[1] 独立行政法人 医薬品医療機器総合機構（PMDA）,「医療用医薬品 添付文書等情報検索」, https://www.pmda.go.jp/PmdaSearch/iyakuSearch/（2026 年 2 月 15 日版データを利用）.

[2] 厚生労働省,「医療用医薬品の添付文書等の記載要領について」（薬生発 0608 第 1 号, 平成 29 年 6 月 8 日）.

[3] 独立行政法人 情報処理推進機構 産業サイバーセキュリティセンター中核人材育成プログラム 7 期生 生成 AI のセキュリティリスクと対策プロジェクト,「テキスト生成 AI の導入・運用ガイドライン」, 2024 年 7 月. https://www.ipa.go.jp/jinzai/ics/core_human_resource/final_project/2024/f55m8k0000003spo-att/f55m8k0000003svn.pdf

[4] P. Manakul, A. Liusie, and M. J. F. Gales, “SelfCheckGPT: Zero-Resource Black-Box Hallu-

cination Detection for Generative Large Language Models," *Proc. EMNLP*, pp. 9004–9017, 2023. https://arxiv.org/pdf/2303.08896

[5] H. Zhang, S. Diao, Y. Lin, Y. Fung, Q. Lian, X. Wang, Y. Chen, H. Ji, and T. Zhang, "R-Tuning: Instructing Large Language Models to Say 'I Don't Know'," *Proc. NAACL-HLT*, pp. 7113–7139, 2024. https://aclanthology.org/2024.naacl-long.394.pdf

[6] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient Memory Management for Large Language Model Serving with PagedAttention," *Proc. 29th ACM Symposium on Operating Systems Principles (SOSP)*, 2023.