一階述語論理を用いた RAG における ハルシネーション判定手法の検討

白木佑弥　西田隼輔　小林優佳　永江尚義
岩田憲治　吉田尚水
株式会社東芝
{yuya.shiraki.k51, shunsuke.nishida.n18,
yuka.kobayashi.f34, hisayoshi.nagae.t34,
kenji.iwata.h13, takami.yoshida.k90}@mail.toshiba@mail.toshiba

概要

近年, 大規模言語モデル（Large Language Model: LLM）は様々なタスクで高い性能を示している一方で, 事実と異なる情報を生成する「ハルシネーション」が課題となっている. RAG (Retrieval-Augmented Generation) においても, LLM が検索文書に基づかない回答をしてしまうことがある. 本研究では, RAG の回答が検索文書に基づくかを一階述語論理で検証する手法を提案する. 検索文書や回答をそのまま扱うだけでは, 正確にハルシネーション判定できないことが多いため, 一文一義となるように文を分割し, 分割後の文を意味や関係性に基づいてグループ化する. 実験の結果, 従来手法と比較してハルシネーションの検知率 (再現率) が 64.5%改善することを確認した.

1　はじめに

近年, 大規模言語モデル（Large Language Model: LLM）は自然言語処理の多様なタスクで高い性能を示し, 企業においても RAG（Retrieval-Augmented Generation）を用いたアプリケーションの活用が進んでいる. RAG はユーザのクエリに関するチャンクを検索文書から検索し, LLM に与えることで, LLM が持たない知識にも回答可能にする仕組みである. 一方, LLM は事実と異なる情報を生成する「ハルシネーション」という問題を抱えており, 特に企業向けの RAG では深刻なリスクとなり得る. 例えば, LLM が生成した誤った情報をユーザが妄信してしまうと, 企業の意思決定や業務に重大な損失を招き得る. このため, RAG の回答がハルシネーションを含むかどうかを検知する技術は不可欠である. 既存のハルシネーション判定手法として RAG の回答にハルシネーションが含まれるか否かを LLM に判定させる手法 [1, 2, 3] がある. しかし, 依然として判定結果や判定に至った根拠にハルシネーションが含まれる可能性があるという根本的な課題を抱えている.

本研究では, RAG の回答が検索したチャンクに基づいているかを一階述語論理 (First-Order Logic: FOL) で判定する手法を提案する. 常識と異なるかどうかを考慮するのではなく, チャンクと異なる情報を回答するハルシネーションを扱う. FOL を用いたハルシネーション判定法には, 論理的なつながりに基づいた推論が可能というメリットがある. それによって, LLM によるハルシネーション判定方法に比べ, 高い説明可能性の実現や判定過程にハルシネーションが含まれることを防ぐことができる.

ただし, FOL で表現するにあたって, 検索したチャンクや RAG の回答には以下の 2 つの問題がある. 1 つ目の問題は, チャンクは複数の内容を含む文章で構成される問題である. そのような文章をそのまま FOL に変換すると, 複雑な FOL になり, ハルシネーションの判定が困難になる. 2 つ目の問題は, チャンクや RAG の回答中の単語の表記や品詞がばらつく問題である. 同義文であっても表記の違いによって FOL の個体変数・述語変数が不一致となることがある. また, 同じ意味の単語が文によって個体変数になったり, 述語変数になったりすることで, 異なる構造の FOL に変換される場合がある. これらの問題によって, 不一致が発生すると, RAG の回答が正しくチャンクに基づいていたとしても, ハルシネーションの判定を正確に行えない.

本研究では, これら問題を解決するために文章を

一文一義となるような文に分割し, 分割後の文を意味や関連性に基づいてグループ化する. 一文一義となるような文に分割することで, 複雑な FOL になることを抑制することができる. また, 同義文をグループ化し, グループ内の文をまとめて FOL に変換することで, FOL の個体変数・述語変数や構造の不一致を抑制する. 本研究では, 独自にハルシネーション有無ラベル付きデータセットを構築し, 提案手法の有効性を検証した.

2 関連研究

自然言語文を FOL に変換する手法として, Yang ら [4] は自然言語文と FOL のペアデータセットを作成し, 自然言語文から FOL への変換タスクで LLaMA2-7B/13B[5] のファインチューニングを行うことで, GPT-4 に匹敵する FOL への変換性能を低コストで実現した. また, 自然言語文の正当性を論理的に判定する手法として, Pan ら [6] は LLM を用いて自然言語文である事実とクエリを FOL に変換し, 自動定理証明器 [7] を用いてクエリが正しいか否かを判定する手法を提案した.

これらの研究は短い文を対象としているが, RAG では長い文を扱う必要がある. 上記のような複数の内容が入り組んだ文をそのまま FOL に変換すると, 複雑な FOL や正確に元の文の情報を捉え切れていない FOL になる可能性が高い. また, 著者らの知る限り, RAG におけるハルシネーション判定に FOL を適用した先行研究は報告されていない.

3 提案手法

本研究では FOL に変換する前に検索したチャンクや RAG の回答のような長い文を一文一義の文に分割する. これにより, 複雑な FOL に変換されることを防ぐ. しかし, チャンクの FOL と RAG の回答の FOL において同じ単語でも異なる表記であったり, 同じ意味の文で異なる FOL の構造であったりすると, ハルシネーションの判定が正確にできない. 本研究では, この問題に対して同じ/似ている意味の文同士をグループ化し, グループ内の文をまとめて FOL に変換する. また, RAG の回答にハルシネーションが含まれるかどうかは, チャンクの FOL から回答の FOL を導けるかで判定する.

提案手法は次の 5 ステップで構成される. ステップ 1 からステップ 3 は LLM で処理を行い, ステップ 4 では自動定理証明器 [7] で処理を行う.

1. ステップ 1: チャンクと RAG の回答をそれぞれ一文一義の文に分割する
2. ステップ 2: 分割後の文を似た意味の文ごとにグループ化する
3. ステップ 3: それぞれのグループの文ごとにまとめて FOL に変換する
4. ステップ 4: 分割後のチャンクを前提, 分割後の RAG の回答を結論として自動定理証明器によって結論が導かれるかを推論する
5. ステップ 5: 全ての結論が前提から導かれる場合, RAG の回答にハルシネーションが含まれないと判定する

これらのステップの概要図を図 1 に示す. それぞれのステップを詳細に説明する.

ステップ 1 チャンクと RAG の回答を一文一義の文に分割する. チャンクや RAG の回答は長い文となる場合が多く, この文をそのまま FOL に変換してしまうと複雑な FOL や誤った FOL に変換されてしまう. そこで, このような長い文を一文一義の文に分割することで, 比較的 FOL に変換しやすい単純で短い文にすることができる.

ステップ 2 ステップ 1 で分割した文を似た意味を持つ文同士でグループ化する. これは似た意味を持つ文は同じ/似た構造の FOL に変換するための処理である. 分割後の文を独立に FOL に変換したり, 分割後の文を全てプロンプトに入力して FOL に変換したりすると似た意味を持つ文でも異なる記号の表記や FOL の構造になってしまう. 似た意味を持つ文をまとめてプロンプトに入力し, それぞれの文を FOL に変換することで表記や構造を統一することができる.

ステップ 3 ステップ 2 でグループ化した文をそれぞれのグループごとに FOL に変換する.

ステップ 4 分割後のチャンクを前提, 分割後の RAG の回答を結論として自動定理証明器に入力し, それぞれの結論が前提から導かれるかどうかを推論する. 一般に分割後の RAG の回答は複数あるため, それぞれの結論ごとに推論を実施する.

ステップ 5 全ての結論を前提から導くことが出来た場合, RAG の回答にハルシネーションが含まれないと判定する.

上記のステップ 1 からステップ 5 までを実施することで, FOL によるハルシネーション判定を行う.

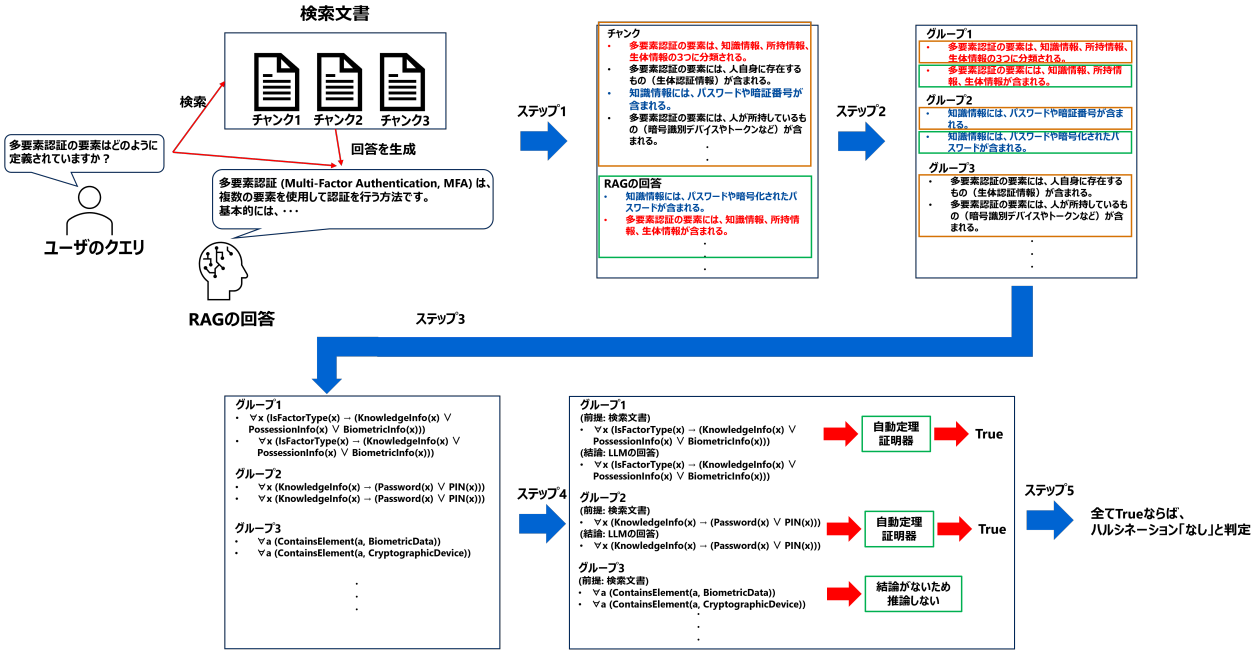

図 1: 提案手法の概要図

4 データセット

FOL を用いた RAG におけるハルシネーション判定を行うために, ハルシネーションの有無がラベルとして付与されているデータセットを作成した. 情報セキュリティに関するサイト[1)] から取得した文書を基にクエリを 68 件作成し, 各クエリに対して 4 つの LLM で回答を生成した (合計 272 件). このうち, 122 件がハルシネーションありのデータである. 作成したデータ例を付録の表 4 に示す.

回答生成時には正解情報を含むチャンクと含まないチャンクを複数与えた. 生成された回答全てに対し, 人手でハルシネーションが含まれるかどうかのラベルを付与した. データセット作成に用いた 4 つの LLM は以下である. 今回はハルシネーションを含む回答の収集を目的としているため, 1B から 14B の比較的小規模モデルサイズの LLM を用いた.

- DeepSeek-R1-Distill-Qwen-14B-Japanese
- gemma2-9B [8]
- llama3.2-1B [9]
- llama3-ELYZA-jp-8B [10]

1) https://www.evaaviation.com/cui-dfars-nist-sp800-171/white-papers/

5 実験

5.1 実験設定

比較手法は, Baseline, Proposed w/o grouping, Proposed の 3 手法である. Baseline は, クエリ・検索したチャンク・RAG の回答を含むプロンプトを入力し, RAG の回答がチャンクに基づくかを判定する. Proposed w/o grouping は, 第 3 章の手順からグループ化を省き, 分割した文を独立に FOL へと変換し, ハルシネーションの判定を行う. Proposed は, 第 3 章の手順に従って判定を行う. Baseline の判定, 提案手法のステップ 1 からステップ 3 の LLM には GPT-4o[11] を用いた.

5.2 実験結果

各手法の正解率, 再現率, F1 値を表 2, 混合行列を表 3 にそれぞれ示す. ただし, Proposed w/o grouping, Proposed では, ステップ 4 でエラーが発生し, ハルシネーションの判定が不能なデータが存在した. そのようなデータを除外したため, 各手法でデータ数は異なる.Baseline は Proposed のデータに合わせて実験を実施した.

表 2 から正解率が Baseline が最も高く, 再現率と F1 値は Proposed w/o grouping が最も高いことがわかる. 表 3 から Baseline はハルシネーションありのデータに対してもハルシネーションなしと判定す

表 1: FOL の比較

検索したチャンク/RAG の回答	Proposed w/o grouping	Proposed
リモートアクセスには、ダイアルアップ、ブロードバンド、ワイヤレスが含まれる。(検索したチャンク)	Include(RemoteAccess, DialUp) ∧ Include(RemoteAccess, Broadband) ∧ Include(RemoteAccess, Wireless)	RemoteAccessType(DialUp) ∧ RemoteAccessType(Broadband) ∧ RemoteAccessType(Wireless)
リモートアクセスにはダイアルアップが含まれる。(RAG の回答)	∀ r ∀ d (RemoteAccess(r) ∧ DialUp(d) → Include(r, d))	RemoteAccessType(DialUp)
リモートアクセスにはブロードバンドが含まれる。(RAG の回答)	∀ r ∀ b (RemoteAccess(r) ∧ Broadband(b) → Includes(r, b))	RemoteAccessType(Broadband)
リモートアクセスにはワイヤレスが含まれる。(RAG の回答)	∀ r ∀ w (RemoteAccess(r) ∧ Wireless(w) → Includes(r, w))	RemoteAccessType(Wireless)
ハルシネーションの判定結果	あり	なし

表 2: 各手法の評価指標

手法	正解率	再現率	F1 値
Baseline	0.68	0.31	0.45
Proposed w/o grouping	0.45	1.0	0.62
Proposed	0.67	0.51	0.58

表 3: 各手法の混合行列

(a) Baseline

	ありと判定	なしと判定
あり	36	79
なし	4	144

(b) Proposed w/o grouping

	ありと判定	なしと判定
あり	114	0
なし	138	0

(c) Proposed

	ありと判定	なしと判定
あり	59	56
なし	30	118

る傾向にある. また, Proposed w/o grouping は全てのデータに対してハルシネーションありと判定している. これは, 同じ意味の単語や文でも FOL の表記や構造が異なるため, 自動定理証明器の推論結果が全て False になることが原因である. 一方, Proposed は Baseline と同程度の正解率を維持しつつ, 再現率と F1 値はそれぞれ 64.5%, 28.9%改善している. これは Proposed がハルシネーションありのデータに対し, 正確に判定できたことに起因する.

5.3 考察

Proposed w/o grouping と Proposed の FOL を比較した結果を表 1 に示す. 表 1 の 2 行目は検索したチャンクと各手法で FOL に変換した結果である. 3-5 行目は RAG の回答と各手法で FOL に変換した結果である. この例では, RAG の回答はチャンクに基づいた回答になっている.

Proposed では回答の FOL のそれぞれがチャンクの FOL の一部となっており, ハルシネーションの判定結果がなしとなっている. 一方, Proposed w/o grouping では RemoteAccess がチャンクでは個体変数, RAG の回答では述語変数として扱われている. また, 1-3 行目は「含まれる」が Include と表現されているのに対し, 4, 5 行目では Includes である. これらによって, Proposed w/o grouping の判定結果はありとなっている. Proposed では, グループ化された文をまとめて FOL に変換するため, 各 FOL の表記や構造が一致するように変換される. 一方, Proposed w/o grouping では各文が独立で FOL に変換されるため, 同じ意味の文でも異なる表現や個体変数と述語変数が統一的に表現されず, ハルシネーションの判定を正確に行えない.

6 おわりに

本研究では, RAG におけるハルシネーション判定手法として, 一階述語論理（FOL）を用いた手法を提案した. 複雑な文章の FOL への変換, FOL の表記や構造の不一致といった問題に対処するために, 提案手法では検索文書と LLM の回答を分割・グループ化を実施した. ハルシネーションの判定は, 変換した FOL と自動定理証明を組み合わせることで実現し, 提案手法の有効性を示した.

一方で, 分割や FOL への変換による情報の欠落や, 処理時間の増加といった課題も残る. 今後はこれらの課題に取り組み, より高精度なハルシネーション判定手法の開発を目指す.

参考文献

[1] Niels Mündler et al. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation, 2024. arXiv preprint arXiv:2305.15852.

[2] Chao Chen et al. Inside: Llms' internal states retain the power of hallucination detection. In **Proceedings of the International Conference on Learning Representations (ICLR)**, 2024.

[3] Weihang Su et al. Unsupervised real-time hallucination detection based on the internal states of large language models. In **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 14379–14391, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Yuan Yang et al. Harnessing the power of large language models for natural language to first-order logic translation. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 6942–6959, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[5] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv preprint arXiv:2307.09288.

[6] Liangming Pan et al. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 3806–3824, Singapore, December 2023. Association for Computational Linguistics.

[7] W. McCune. Prover9 and mace4. http://www.cs.unm.edu/~mccune/prover9/, 2005–2010.

[8] Gemma Team. Gemma. 2024. 最終アクセス: 2026 年 1 月 7 日.

[9] Meta AI. Llama3.2-1b-instruct, 2024. 最終アクセス: 2026 年 1 月 7 日.

[10] Masato Hirakawa et al. elyza/llama-3-elyza-jp-8b, 2024. 最終アクセス: 2026 年 1 月 7 日.

[11] OpenAI et al. Gpt-4o system card, 2024. arXiv preprint arXiv:2410.21276.

商標について

"GPT-4o"及び"LLaMA"はそれぞれ OpenAI 社, Meta 社の登録商標です. また, その他本校に掲載の商品, 機能などの名称は, それぞれ各社が商標として使用している場合があります.

付録 (Appendix)

表 4: 構築したデータセットの一例

クエリ	監査ログ取得機能の管理を特権ユーザーの一部の者に限定する目的は何ですか？
RAG の回答	監査ログ取得機能の管理を特権ユーザーの一部の者に限定する目的は、以下の通りです。 1. **監査情報の信頼性確保** 特権アクセスを持つ個人が監査活動を妨害したり、監査記録を改ざんすることでシステムのセキュリティ状態が損なわれるリスクを軽減します。これにより、監査データの正確性と整合性を維持できます。 2. **責任範囲の明確化** 監査関連の特権を持つユーザーのみに限定することで、システムコンポーネントごとの説明責任（accountability）を明確にします。これにより、変更や操作が追跡可能になります。 3. **セキュリティ要件の遵守** NIST-SP-800-171-Rev.2 の要件「3.3.9 監査ログ取得機能の管理を特権ユーザーの一部の者に限定する」を満たすためです。これにより、システムのセキュリティ態勢が脆弱性から保護されます。 4. **内部監査の効果性向上** 特権アクセスを持つ個人が監査活動に関与することで、潜在的な不正や誤操作を早期に検出・防止できます。これにより、組織全体のセキュリティガバナンスが強化されます。 この措置は、システムコンポーネントごとの異なる時間粒度（例：数百ミリ秒以内）での同期管理（3.3.7）や、監査情報の物理的保護（3.8）など、他のセキュリティ要件と連動して効果を発揮します。
ハルシネーションの有無	あり
ハルシネーションの箇所	「これにより、監査データの正確性・・・」から「・・・他のセキュリティ要件と連動して効果を発揮します。」まで
ハルシネーションの説明	検索結果にないことを説明している