

LLM の生成テキストの真偽検証のための 日本語言説分解データセットの構築

政野 美和（一橋大学ソーシャル・データサイエンス学部
／国立情報学研究所大規模言語モデル研究開発センター）*

櫛 リベカ（東京工科大学コンピュータサイエンス学部）

櫛 惇志（一橋大学ソーシャル・データサイエンス研究科）

清丸 寛一（国立情報学研究所大規模言語モデル研究開発センター）

中山 功太（国立情報学研究所大規模言語モデル研究開発センター）

堀尾 海斗（早稲田大学理工学術院）

源 怜維（早稲田大学理工学術院／国立情報学研究所大規模言語モデル研究開発センター）

橘 秀幸（国立情報学研究所大規模言語モデル研究開発センター）

河原 大輔（早稲田大学理工学術院／国立情報学研究所大規模言語モデル研究開発センター）

要旨

大規模言語モデル (LLM) の生成テキストは誤情報（ハルシネーション）を含むことがあるため、情報の真偽を検証する真偽検証システムの開発が期待されている。現在主流の真偽検証の方式では、まず生成テキストを独立した最小粒度の情報である言説 (claim) に分解する。次に、言説をクエリとして根拠文を検索し、得られた言説-根拠文ペアに対して判定予測を行う。言説単位での真偽検証を行うことで、真偽検証結果に対する説明性が向上する。その際、高精度な真偽検証のためには適切な粒度に言説が分解されていることが求められる。これらを踏まえて、本研究では日本語言説分解データセットの構築に取り組む。本発表では、データセット構築の中間報告として、言説分解のガイドラインの設計について報告する。ガイドライン設計においては、抽出される言説が真偽検証において有用な形式となることを目的としつつ、作業結果の揺れを軽減する言説分解ルールを検討した。作成された言説分解ガイドラインを用いて試験的な言説分解を実施したところ、複雑な構造を持つ生成テキストに対しても複数の作業員間で揺れのない分解を行うことができたケースが観測された。その一方で、並列構造の解釈、焦点語の選択、作業員の知識などの差異によって作業員間で揺れが生じた分解も存在した。

1. はじめに

大規模言語モデル (Large Language Model ; LLM) を基盤とする AI チャットボットの普及が急速に進んでいる。LLM は、大量のテキストデータで学習された、高度なテキスト理解と流暢なテキスト生成が可能な機械学習モデルである。現在、LLM の汎用的なテキスト処理能力を活用してユーザの様々な問いかけや指示に柔軟に応えるチャットボットが急速に普及し

* 5123053k@g.hit-u.ac.jp

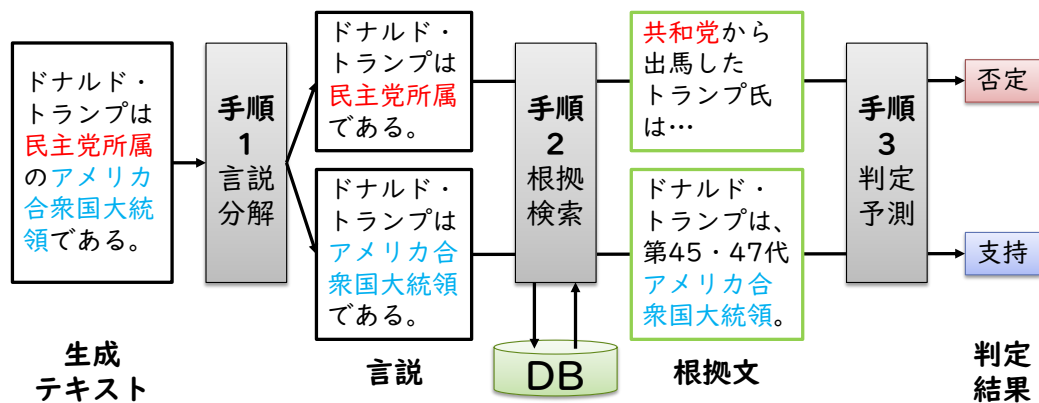


図1 真偽検証システムの概要図

ている (Hu 2023). 最近は医療分野といった高い信頼性が求められる領域への応用も本格化しており (Singhal et al. 2025), LLM とその応用システムが社会に与える影響は日増しに大きくなっている。

このように急速に普及が進んでいる LLM の欠点の一つとして、その生成テキストには誤情報であるハルシネーション (hallucination) が含まれる場合があることが挙げられる。学習用コーパスに正しい情報が含まれていた場合にもハルシネーションが起こることが明らかになっている (Huang et al. 2025, Xu et al. 2024) ため、ハルシネーションを未然に防ぐのは容易ではない。

こうした背景から、LLM の出力テキストに対して、情報の真偽を検証するシステム (真偽検証システム) の研究・開発が進んでいる (Guo et al. 2022, Zeng et al. 2021, Kotonya and Toni 2020, Kamoi et al. 2023, Wang et al. 2024). これにより、LLM の信頼性が高まり、現在はまだ LLM の利用が進んでいない領域にも今後 LLM がより浸透していくことが期待される。さらに、真偽検証によって検出された否定根拠とその修正内容をフィードバックとして LLM に与えて応答を再生成することで、生成テキストのハルシネーションが軽減されることが確認されている (Lei et al. 2023). このことから、真偽検証は生成テキストの品質改善にも寄与する技術である。

現在主流の真偽検証システムは下記の 3 ステップから構成される。

1. 言説分解 (claim decomposition)/言説検出 (claim detection)
2. 根拠検索 (evidence retrieval)
3. 判定予測 (verdict prediction)

概要図を図 1 に示す。

まず、手順 1. 言説分解では、入力テキスト (本研究においては LLM の生成テキスト) を、一つの物事に関する性質や関係を表す独立した最小粒度の情報単位である言説 (claim)⁽¹⁾ に分

⁽¹⁾ 文献によっては一纏まりのテキストを「言説 (claim)」, 言説から分解される独立した最小粒度の情報を「副言説 (sub-claim)」と呼称する場合もある。本研究では先行研究である Factcheck-Bench (Wang et al. 2024) に準拠し、独立した最小粒度の情報単位を「言説」と呼ぶ。

解する。例えば、「ドナルド・トランプは民主党所属のアメリカ合衆国大統領である。」という生成テキストは「ドナルド・トランプは民主党所属である。」と「ドナルド・トランプはアメリカ合衆国大統領である。」に分解される。次に、手順 2. 根拠検索では、言説をクエリとしてデータベースに対して検索を行い、根拠文（パッセージ）を取得する。最後に、手順 3. 判定予測では、取得された言説-根拠文のペアに対して自然言語推論 (Natural Language Inference; NLI) を行うことで、根拠文が言説の内容を支持する（含意する）か判定する。

言説という細かい粒度の情報に対して真偽検証を行うことで、真偽検証結果に対する説明性が向上する。例えば、言説分解前の「ドナルド・トランプは民主党所属のアメリカ合衆国大統領である。」が真偽検証の結果事実ではないと判定された場合には、「所属政党」が誤っていたのか「役職名」が誤っていたのか不明である。それに対して、言説単位で真偽検証を行うことで、細かい粒度での誤情報を検出することができる。ただし、言説への分解結果が誤っていた場合には、検証する情報の単位が実態に則さず、適切な判定予測を行うことができない。そのため、言説分解の品質は真偽検証の性能に大きな影響を与えることが報告されている (Wanner et al. 2024)。以上より、高品質な言説分解は高性能な真偽検証システムの開発において非常に重要である。

Kamoi et al. (2023), Wang et al. (2024) では LLM の文脈内学習 (In-Context Learning ; ICL) を用いて言説分解を行っている。ICL は、解きたいタスクの説明と入力を LLM の文脈（プロンプト）に与え、LLM に回答を生成させる手法である。ICL による言説分解では、タスク説明の一部として与える言説分解の具体例（few-shot 事例）が出力の品質を左右する主要因となる (Wanner et al. 2024)。我々は先行研究で英語を対象に有効性が確認された few-shot 事例を含むプロンプトを日本語化し、日本語生成テキストの言説分解を行う予備実験を実施した。その結果、言説は必ずしも適切な粒度で得られず、しばしば過剰な分解や不十分な分解に陥った。これは、英語以外の言語における LLM の性能・頑健性の低下 (Li et al. 2025, Gan and Mori 2023, Sakai et al. 2024) が原因と考えられ、日本語の適切な言説分解を実現するには few-shot プロンプトの改善を含む手法の改良やモデルのチューニングが必要であることが確認された。

本研究では、日本語の真偽検証システムにおける言説分解のステップの改善のため、生成テキストとその言説セットが含まれる言説分解データセットの構築を行う。言説分解データセットの開発により、言説分解手法の定量評価、言説分解に存在する課題を明らかにするための分析、評価と分析に基づく言説分解手法の改善（高品質な few-shot 事例やデータセット構築の副産物である言説分解ガイドラインを用いたプロンプトの改良、言説分解に特化したモデルへのファインチューニングなど）が可能となる。

本論文では、言説分解データセット構築の中間報告として、言説分解のガイドラインの設計について報告する。ガイドライン設計においては、得られた言説が真偽検証において有用な形式となることを目的とした。具体的には、後段の根拠検索における検出力を向上させることや、判定予測における曖昧性を排除するためのルールを設定した。また、作業結果の揺れを軽減する言説分解ルールも検討した。

以降、本論文では、2 節にて関連研究を紹介する。その後 3 節にて言説分解データセットの

設計について、4 節にて言説分解について述べる。その後 5 節にて初期分析結果について報告を行い、6 節にて本研究を纏める。

なお本論文は LLM の生成テキストの真偽検証を最終的な目的とするという性質上、本文で扱う例文の中には事実と異なる内容のものが含まれることに留意されたい。

2. 関連研究

2.1 真偽検証システム

真偽検証に関する研究は従来からフェイクニュースなどを対象として取り組まれていた (Guo et al. 2022, Zeng et al. 2021, Kotonya and Toni 2020)。その後、真偽検証のための代表的なデータセットである FEVER (Thorne et al. 2018) やその後続のデータセット (Thorne et al. 2019, Aly et al. 2021, Schlichtkrull et al. 2023) が登場して、真偽検証システムの研究を牽引している。なお、これらのデータセットでは、判定予測のラベルは、支持 (supported)、否定 (refuted)、不明 (not enough information) が採用されている。

WiCE (Kamoi et al. 2023) は言説単位の判定予測ラベルに加えて、トークン単位の判定予測ラベルも付与しているため、より細かい粒度での真偽検証が可能となる。言説分解は GPT-3 を用いて自動化している。また、WiCE の判定予測ラベルとしては、支持 (supported)、部分支持 (partially-supported)、不支持 (not-supported) が用いられている。

2.2 LLM の真偽検証フレームワーク

LLM の生成テキストを対象とした真偽検証のためのフレームワークである Factcheck-Bench (Wang et al. 2024) は、LLM の生成テキストの真偽検証のために、前述の真偽検証システムより細かいステップから構成される真偽検証フレームワークの提案とベンチマークの構築を行っている。なお、このフレームワークは、真偽検証結果を踏まえた応答再生成によって、生成テキストの改善も行っている。具体的なステップは下記の通りである。

1. 言説分解 (decompose)
2. 文脈依存の排除 (decontextualise)
3. 検証対象言説の特定 (identify checkworthy claims)
4. 根拠検索 (retrieve and collect evidence)
5. 判定予測 (identify stance of evidence)
6. 修正対象言説の特定 (determine correction)
7. 言説修正 (edit claims)
8. 生成テキスト修正 (revise response)

1. 言説分解

本ステップでは、GPT-3.5 Turbo を用いて言説に分解する。

2. 文脈依存の排除

次のステップでは、言説分解によって失われた文脈情報を補間することで、言説を解釈するうえで前後の言説を参照する必要がなくなる。具体例を挙げて説明すると、1. 言説分解の結果、ある文から 2 つの言説「ドナルド・トランプはアメリカ合衆国大統領である。」「彼は民主党所

属である。」に分解されたとする。このときに、文脈依存の排除のステップでは、言説「彼は民主党所属である。」を「ドナルド・トランプは民主党所属である。」に修正する。

3. 検証対象言説の特定

ここでは検証価値のある言説を特定する。このステップでは、各言説は下記の 4 種類のカテゴリに分類される。

- 真偽検証可能な言説 (factual claim)
- 意見 (opinion)
- 非言説 (not a claim)：疑問 (questions), 感嘆 (exclamations), 命令 (imperatives)
- その他 (others)：例「その質問には回答できません」

意見・非言説・その他は真偽検証を行う意義がないため、続く 4. 根拠検索以降のステップでは、真偽検証可能な言説のみを対象とする。これにより、計算リソースや否定根拠の誤検出を削減する。

4. 根拠検索

根拠検索では、言説からクエリを作成して Google 検索を行い、上位 5 件の根拠文を取得する。

5. 判定予測

判定予測では、言説-根拠文のペアに対して支持 (support), 部分支持 (partially support), 否定 (refute), 無関係 (irrelevant) の 4 種類のラベルのいずれかを判定する。

6. 修正対象言説の特定, 7. 言説修正, 8. 生成テキスト修正

これらのステップは、生成テキストの改良のためのステップである。ステップ 6, 7 にて言説の修正案を作成して、ステップ 8 にて LLM ヘフィードバックを与える。なお、ステップ 6-8 は、与えられたコンテキストに対して NLI を行い、その結果を踏まえて LLM に応答を再生成させることでハルシネーションを軽減させた手法である CoNLI (Lei et al. 2023) と同等の処理を行っていると思なすことができる。従って、真偽検証システムとその拡張としての改良のフィードバックによって、生成テキストの品質向上に寄与することが期待される。

2.3 ハルシネーションの軽減

LLM のハルシネーションの軽減手法には、学習データの品質改善 (Albalak et al. 2024), デコーディングの調整 (Li et al. 2023), モデルによる自己検証 (Manakul et al. 2023, Zhang et al. 2024), 真偽検証結果を踏まえた再生成 (Wang et al. 2024), 検索拡張生成 (Retrieval-Augmented Generation; RAG) (Lewis et al. 2020, Gao et al. 2023) などがある。中でも、RAG は真偽検証システムと同様に検索技術を用いる技術である。

LLM に検索モジュールを組み込むことで知識を補完する RAG によってタスク遂行性能が向上することが多数報告されている (Lewis et al. 2020, Gao et al. 2023)。さらに、RAG によって高品質な外部情報を利用することでハルシネーションが減少することも報告されている (Tonmoy et al. 2024, Lewis et al. 2024, Zhang and Zhang 2025, Magesh et al. 2025, Chu et al. 2025)。さらに、RAG では LLM の応答生成時にハルシネーション抑制や生成テキストの品質改善を行うため、真偽検証システムのように応答生成後に後処理でハルシネーション検出や生成テキストの品質改善を行うアプローチよりも効率的であるといえる。

しかしながら、RAG を用いてもハルシネーションを完全に抑制することは容易ではない。例えば、法律ドメインにおける RAG の活用に関する研究 (Magesh et al. 2025) では、RAG を用いることでハルシネーションが軽減されるものの完全には排除されないことを報告している。そのため、法律分野において AI を活用するうえでは専門家による LLM の生成テキストの検証の責任について言及している。

これらを踏まえて、RAG は LLM の生成テキストのハルシネーション軽減に貢献を果たしているものの、真偽検証システムを併用することでより高品質かつ信頼性の高い利用をもたらすと考えられる。

2.4 言説分解

Wanner et al. (2024) は、言説を粒度とした事実性に関する評価タスク（生成テキストの事実精度評価 (Min et al. 2023)、含意関係認識 (Chen et al. 2023)、真偽検証 (Kamoi et al. 2023, Wang et al. 2024)) において、言説分解の品質は各タスクの性能に大きな影響を与えることを実験的に確認した。これは、言説分解によって言説の個数や個々の言説の持つ情報が決まるため、言説分解におけるエラーは事実性評価にも影響を及ぼすためである。なお、Wanner et al. (2024) は、高品質な言説分解には、網羅性 (coverage)、整合性 (coherence)、原子性 (atomicity) を考慮する必要があると主張している。網羅性は分解で作成された言説が全体として元の生成テキストのすべての部分を含むこと、整合性は分解で作成された個々の言説が元の生成テキストに忠実であること（含意されること）、原子性は言説が可能な限り原子的である（粒度が細かい）ことを表す。

また、FactLens (Mitra et al. 2024) では、言説の品質の評価のために下記の 6 種類の指標を考慮している。

1. 原子性 (atomicity)：言説が可能な限り原子的であること
2. 十分性 (sufficiency)：言説に曖昧性がなく個々の言説を独立に真偽検証できること
3. 捏造 (fabrication)：元の生成テキストに含まれない情報を追加していないこと
4. 網羅性 (coverage)：分解で作成された言説が全体として元の生成テキストのすべての部分を含むこと
5. 冗長性 (redundancy)：言説間で情報が重複していないこと
6. 可読性 (readability)：ユーザにとって読みやすいこと

このうち、主要な指標は 1. 原子性、2. 十分性、3. 捏造、4. 網羅性であり、5. 冗長性と 6. 可読性は検証可能性や精度に直接影響を与えるわけではないものの、あれば望ましい指標として扱っている。

3. 言説分解データセットの設計

本研究で構築する言説分解データセットは、(i) LLM の生成テキスト、(ii) 生成テキストから言説分解により作成された言説、(iii) 個々の言説に対する言説分類のラベルから構成される。

以下は生成テキスト (1) に対する 2 つの言説 (2) と、個々の言説の分類ラベルの例である。

(1) 日本の初代内閣総理大臣は伊藤博文です。帝国憲法の制定に尽力しました。

(2) 1. 伊藤博文は日本の初代内閣総理大臣です。言説分類ラベル：Check-worthy

2. 伊藤博文は帝国憲法の制定に尽力しました。言説分類ラベル：Check-worthy

本データセット構築には、言説分解と言説分類という二つの作業が含まれる。二つの作業の目的は、後段のステップである根拠検索と判定予測の際に望ましい形の言説を抽出することである。

3.1 言説分解

先行研究を踏まえ、言説は独立した最小粒度の情報単位であり、各言説は一つの物事に関する性質や関係を表すものと定義した。そのため生成テキストに対する言説分解のステップでは、以下の三つの原則にしたがって、元の生成テキストと意味が変わらない範囲、すなわち言説が元の生成テキストから含意される範囲で、可能な限り細かい粒度に分解する。言説分解の目的は、根拠検索と判定予測の際の精度向上であるため、言説は独立した最小粒度の情報を含む。

- 原則 1：各言説は前後の文脈なしに解釈できるようにする⁽²⁾
- 原則 2：分解された言説から元の情報を復元できるようにする
- 原則 3：なるべく元のテキストと同じ表現を用いるようにする

3.2 言説の分類

このように言説分解して得られた言説は、すべてが同等に真偽検証に値する言説であるとは限らない。例えば個人の主観については検証は困難であるし、疑問文などのようにそれ自体の真偽が定義困難なものもある。そこで、言説分解後、真偽検証可能な言説のみに対して真偽検証を行うために、分解した言説を以下の 2 つのカテゴリに分類する。

- カテゴリ 1：Check-worthy 真偽検証に値する
 - － 客観的な事実や客観的な評価
 - － 例) この映画の上映時間は 2 時間 30 分だ。
 - － 例) この映画は多くの人に評価されている。
- カテゴリ 2：Not check-worthy 真偽検証に値しない
 - － 主観的な意見や個人の経験
 - － 例) この映画はとても感動的だ。
 - － 例) 彼はこの映画を 3 回鑑賞した。

なお、本論文は中間報告のため言説分解にフォーカスし、言説分類の詳細なガイドラインについては省略する。

4. 言説分解のガイドライン

3.1 節の言説分解の品質が真偽検証の精度に影響することから、高品質な言説分解データセットの構築が求められる。それにはまず、3.1 節で述べた原則 1 から原則 3 の条件を満たし

⁽²⁾ Factcheck-Bench (Wang et al. 2024) では言説分解後に文脈依存の排除を行っていたが、本研究では言説分解時に行うこととする。

た、独立した最小粒度の情報を含む言説への分解を実現することが必要である。それに加え、一貫性のある信頼性の高いデータセットを構築するためには、作業者間の揺れを抑えることが重要である。さらに、言説が根拠検索の精度をより高める表現であること、また、判定予測しやすい表現であることも考慮する必要がある。本節では、これらの側面から言説分解のガイドラインの設計について述べる。

4.1 分解の粒度

独立した最小粒度の情報単位である言説を作成する（2.4 節「十分性」と対応）ために、言説分解では生成テキストをできるだけ細かい粒度に分解することが基本である（2.4 節「原子性」と対応）。これにより、文や節の境界、並列構造や関係節は分解の対象となる。一方、言説分解は元の生成テキストと意味が変わらない範囲（すなわち言説が元の生成テキストから含意される範囲）で行う必要がある（2.4 節「整合性」「捏造」と対応）。そのため、構造的には分解可能な言説であっても、意味的にはそれ以上分解できない場合がある。以下に代表的な 2 事例を挙げる。なお、以降のルールはいずれも原則 2「分解された言説から元の情報を復元できるようにする」（2.4 節「網羅性」と対応）に留意して設計した。

4.1.1 並列構造

先述の通り、言説分解では「A や B」「A と B」といった並列構造は原則として個別の要素に分解する。たとえば以下のテキスト (3) は、(4) の 2 つの言説に分解される。

(3) テキスト：

アメリカ五大湖から大西洋にいたる地域はかつては自動車製造や重工業で発展しました。

(4) OK な分解：

1. アメリカ五大湖から大西洋にいたる地域はかつては自動車製造で発展しました。
2. アメリカ五大湖から大西洋にいたる地域はかつては重工業で発展しました。

上の言説分解では「自動車製造や重工業で」という並列構造を分けてそれぞれを含む言説を作成している。

一方で、分解しないのが適切であると考えられる例外的な並列構造も存在する。たとえば以下のテキスト (5) の下線部では 4 つの作品名が並列されており、並列構造を分解するという原則に則ると、太字部分に対して作成される言説は (6) となる。

(5) テキスト：

『ラインの黄金』『ワルキューレ』『ジークフリート』『神々の黄昏』はそれぞれ独立した作品でありながら、**一つの壮大な物語を形成しています。**

(6) NG な分解：

1. 『ラインの黄金』は一つの壮大な物語を形成しています。
2. 『ワルキューレ』は一つの壮大な物語を形成しています。
3. 『ジークフリート』は一つの壮大な物語を形成しています。
4. 『神々の黄昏』は一つの壮大な物語を形成しています。

しかしテキスト (5) が意味するのは 4 作品が一つの壮大な物語を形成しているということであ

り、並列構造を (6) のように分解することによって、本来のテキストの意味が変化してしまう。このように、元の生成テキストがもつ意味を成立させる上で、並列された要素が同時に存在することが重要だと考えられる場合には、並列構造は分解しないものとする。したがって、ここでは並列構造を含む (7) の言説を採用する。

(7) OK な分解:

1. 『ラインの黄金』『ワルキューレ』『ジークフリート』『神々の黄昏』は一つの壮大な物語を形成しています。

4.1.2 制約・条件・限定の表現

修飾表現と被修飾表現は言説分解時に切り離すのが原則であり、以下のテキスト (8) は (9) のように言説分解される。

(8) テキスト:

天平文化は中国・唐の影響を強く受けた貴族文化です。

(9) OK な分解:

1. 天平文化は中国・唐の影響を強く受けています。
2. 天平文化は貴族文化です。

ただし、制約・条件・限定の表現については切り離すことで言説の意味が変わったり、曖昧性が生じたりする場合がある。このようなケースでは、必要に応じて個々の言説に制約・条件・限定の表現を含める。たとえば、以下のテキスト (10) に対する言説分解は、(11) でなく、(12) であるとする。

(10) テキスト:

日本のテレビ業界で「プライムタイム」とは、一般的にゴールデンタイムと呼ばれる 19 時から 23 時の時間帯のことを指します。

(11) NG な分解:

1. 「プライムタイム」は、一般的にゴールデンタイムと呼ばれます。
2. 「プライムタイム」は、19 時から 23 時の時間帯のことを指します。
3. 「プライムタイム」は日本のテレビ業界の用語です。

(12) OK な分解:

1. 日本のテレビ業界で「プライムタイム」とは、一般的にゴールデンタイムと呼ばれる。
2. 日本のテレビ業界で「プライムタイム」とは、19 時から 23 時の時間帯のことを指します。

上記の望ましくない分解 (11) では、「日本のテレビ業界で」という限定の表現は言説 (11-3) に含まれており、分解された言説から元の情報を復元することができる。すなわち、原則 2 (3.1 節) は満たされている。しかし、「日本のテレビ業界で」という前置きを切り離れた言説 (11-1), 言説 (11-2) では言説の意味が変わったり曖昧性が生じる恐れがある。たとえば、日本のテレビ業界におけるプライムタイムは 19 時から 23 時の時間帯であるが、プライムタイムが指す時間帯は国によって異なる。したがって元のテキスト (10) が真であっても言説 (11-2) は必ずしも真であると言い切れず、生成テキストと言説の含意関係が成り立たなくなる。この

ように、修飾表現を切り離すことで生成テキストの意味が大きく変わるおそれがある場合は制約・条件・限定の表現であるとみなし、それ以上分解しない。

同様に、テキスト (13) の太字部分から作成する言説は、(14) のように 2 つの言説に分解されるのではなく、(15) のようになる。

(13) テキスト：

デリシャスとゴールデンデリシャスを掛け合わせて誕生した**世界最大の大きさを誇るリンゴの品種は「ふじ」**です。

(14) NG な分解：

1. 「ふじ」はリンゴの品種です。
2. 「ふじ」は世界最大の大きさを誇ります。

(15) OK な分解：

1. 「ふじ」は世界最大の大きさを誇るリンゴの品種です。

望ましくない分解 (14) では、元のテキスト中の「リンゴの品種」「世界最大の大きさを誇る」に対応する言説が作成されており、各言説が最小粒度の情報を持つように分解されている。しかし、「リンゴの品種」という表現を切り離したことによって、言説 (14-2) を「『ふじ』は（あらゆるものの中で）世界最大の大きさを誇る」と解釈することも可能となり、元のテキストにはない解釈の曖昧性が生じてしまうため、この分解は採用しない。

4.2 言説の表現

言説の適切な粒度が定まっても、言説分解によって作成される言説が一意に定まるわけではない。たとえば、以下の (16) の太字部分から言説を作成する際には、「この時間帯」をどの表現に置き換えて文脈依存性を解消するかによって、少なくとも (17) に示す複数の候補がありうる。

(16) テキスト：

日本のテレビ業界で「プライムタイム」とは、一般的にゴールデンタイムと呼ばれる 19 時から 23 時の時間帯のことを指します。**この時間帯は視聴者の年齢層が幅広い**ため、視聴率が高くなる傾向があります。

(17) 言説の候補

- a. 日本のテレビ業界で**プライムタイム**は視聴者の年齢層が幅広い。
- b. 日本のテレビ業界で**ゴールデンタイム**は視聴者の年齢層が幅広い。
- c. 日本のテレビ業界で**19 時から 23 時の時間帯**は視聴者の年齢層が幅広い。

「プライムタイム」はゴールデンタイムと呼ばれ、ゴールデンタイムは 19 時から 23 時の時間帯を指すため、(17) の下線部はすべて同じ事柄を指している。元のテキスト (16) の文構造からも「この時間帯」の先行詞は自明ではなく、(17) のバリエーションは作業者間の揺れとなりうる。

同様に、テキスト (18) から「東大寺正倉院」を含む言説を作成すると、(19) に示す複数の候補がありうる。

(18) テキスト：

聖武天皇の時代に栄えた、**東大寺正倉院**や唐招提寺金堂など、中国・唐の影響を強く受けた貴族文化は「天平文化」と呼ばれます。

(19) 言説の候補

- a. 東大寺正倉院は中国・唐の影響を強く受けた。
- b. 東大寺正倉院は「天平文化」のものです。

この例では、(19) に示す言説はいずれも元のテキストから含意すると考えられるが、「東大寺正倉院や唐招提寺金堂など」の直接の係り先が明らかでないため、(19) の 2 つの言説のいずれが優勢であるか自明でなく、作業者間の揺れとなりうる。

このように、意味的に同等でありながら表現が異なる複数の言説から 1 つを決定し、作業者間の揺れを軽減するために、「焦点語」という概念を導入する。生成テキストの文やその一部において主題とみなせ、検索における検出力が高いことが期待される表現のことを「焦点語」と呼ぶ。先の例においては、以下に示すように下線部を焦点語とみなせる。

(20) テキスト：

聖武天皇の時代に栄えた、東大寺正倉院や唐招提寺金堂など、中国・唐の影響を強く受けた貴族文化は「天平文化」と呼ばれます。

(21) テキスト：

日本のテレビ業界で「プライムタイム」とは、一般的にゴールデンタイムと呼ばれる 19 時から 23 時の時間帯のことを指します。この時間帯は視聴者の年齢層が幅広いため、視聴率が高くなる傾向があります。

以下の例では、第一節の焦点語は「P 型半導体」、第二節の焦点語は「N 型半導体」とみなせる。

(22) テキスト：

ホールが使われるのをP 型半導体というのに対し、自由電子が使われるのをN 型半導体といいます。

文あるいはその一部を言説に分解する際は、まず焦点語を決定し、焦点語を中心に分解する。作成する言説においては焦点語を主語とする語順を基本とするが、日本語としての自然さを優先する。以下はテキスト (20) を、焦点語「天平文化」を中心に分解して作成した言説である。

(23) OK な分解：

1. 「天平文化」は聖武天皇の時代に栄えました。
2. 東大寺正倉院は「天平文化」のものです。
3. 唐招提寺金堂は「天平文化」のものです。
4. 「天平文化」は中国・唐の影響を強く受けました。
5. 「天平文化」は貴族文化です。

このように焦点語を中心に言説分解を行うことで、作成する言説の候補が絞られ、作業者間の揺れの軽減に役立つ。加えて、複数存在しうる言説の候補のうち、検索における検出力が高いことが期待される表現を含んだ言説、すなわち後段の根拠検索において有用な言説を選択することが可能となる。

4.3 保存しない情報

言説分解の原則 2 として「分解された言説から元の情報を復元できるようにする」という点を挙げているが (3.1 節参照)、以下のような情報は言説分解時に保存せず、細かく分解することを優先する。

4.3.1 節と節の関係 (談話構造)

節と節の関係 (談話構造) は、言説よりも大きな単位で構成されるため、言説分解によって失われる場合がある。したがって、節と節の関係を保存することを重視すると、最小粒度の情報を含む言説に分解することが困難になる可能性がある。

たとえば、テキスト (24) の太字に関して、接続詞「ものの」による節間関係を保持した言説分解を行うと (25) のようになる。ここではより細かい粒度の言説を作成することを優先し、(26) に示す言説分解を採用する。

(24) テキスト：

かつては自動車製造や重工業で発展したものの、近年衰退した、アメリカ五大湖から大西洋にいたる地域のことを、「さび付いた工業地帯」という意味の英語で「Rust Belt(ラストベルト)」といいます。

(25) NG な分解：

1. 「Rust Belt(ラストベルト)」はかつては自動車製造で発展したものの、近年衰退した。
2. 「Rust Belt(ラストベルト)」はかつては重工業で発展したものの、近年衰退した。

(26) OK な分解：

1. 「Rust Belt(ラストベルト)」はかつては自動車製造で発展した。
2. 「Rust Belt(ラストベルト)」はかつては重工業で発展した。
3. 「Rust Belt(ラストベルト)」は近年衰退した。

4.3.2 「など」「始めとする」などの例を挙げる表現

一般に、対象 A に関する「A など」「A を始めとする」という表現は、「A の他にも存在する」という意味を含んでいる。このような例を挙げる表現を含む言説に対して判定予測を行う際に、A を含む根拠文の中に必ずしも例を挙げる表現が含まれているとは限らない。例を挙げる表現は「他にも存在する」という意味を含むため、例を挙げる表現を含む言説と含まない根拠文の判定予測関係は支持と判定されない。その一方で、判定予測は個々の言説-根拠文ペアに対して独立に行われるため、この言説では A についてのみ真偽検証すれば十分であると考えられる。そのため、例を挙げる表現については今回のデータセットでは扱わないこととする。これにより、テキスト (27) の太字部分からは、(28) ではなく (29) の言説が作成される。

(27) テキスト：

企業買収の際などに行われる「株式公開買付」のことをアルファベット 3 文字で表すと「TOB」となります。

(28) NG な分解：

1. 「株式公開買付」は企業買収の際などに行われる。

(29) OK な分解:

1. 「株式公開買付」は企業買収の際に行われる。

4.3.3 数え上げに関する情報

「A に当てはまるのは B と C です」という表現からは「B は A です」「C は A です」という分解のみを作成し、元の表現から読みとれる「A に当てはまるのは 2 つだ」という情報は言説として作成しない。

たとえば (30) の太字部分からは、すべての桁の数字が 1 である数字の呼び方は 2 つあるように読み取れ、(31) の分解ではその情報を 3 番目の言説として作成している。

(30) テキスト:

「repeated unit」を略して名付けられた、**すべての桁の数字が 1 である数のことを「unit digit number」または「repeating unit number」と言います。**例えば、111 や 1111 などがこれに該当します。

(31) NG な分解:

1. すべての桁の数字が 1 である数のことを「unit digit number」と言います。
2. すべての桁の数字が 1 である数のことを「repeating unit number」と言います。
3. すべての桁の数字が 1 である数の呼び方は二つあります。

しかし、(31-3) のような言説を作成することにはいくつか問題がある。まず、元のテキスト (30) が意味するのは、「すべての桁の数字が 1 である数の呼び方が少なくとも二つある」ということに留まるが、作業者が「すべての桁の数字が 1 である数の呼び方は二つだけある」と読み取る懸念がある。そのため、作業者が (31-3) のように「二つ」という表現で言説を作成した場合、その意図が「少なくとも二つ」なのか「二つだけ」なのかは明らかでない。これを防ぐには、作業者が尺度推意と含意を区別する必要があるが、現実的ではない。さらに、「少なくとも二つある」という言説を作成したとしても、後段の根拠検索・判定予測において呼び方が一つしか発見できなかった場合に、判定を「部分支持」と予測すべきか「矛盾」と予測すべきかが自明ではないという問題もある。

これらの問題を考慮し、(31-3) の言説は作成しない方針とし、望ましい分解結果は (32) であると定める。

(32) OK な分解:

1. すべての桁の数字が 1 である数のことを「unit digit number」と言います。
2. すべての桁の数字が 1 である数のことを「repeating unit number」と言います。

5. 初期の言説分解の分析

作成した言説分解ガイドラインにしたがって、試験的な言説分解を行った。以降、言説分解が一致したケースと言説分解の判断が難しかったケースを報告する。

言説分解の対象となる生成テキストは、オープンドメイン質問応答タスクのデータセットである AI 王 Version 2.0 開発用データ⁽³⁾に含まれる 1,000 件のクイズ問題に対する解答を

⁽³⁾ <https://sites.google.com/view/project-aio/dataset#h.lysjxtu9gi8e>

表 1 クイズ問題と生成テキストの例

クイズ	氷った海に穴を開けて漁をすることから、漢字で「氷の下の魚」と書くタラ科の魚は何？
生成テキスト	漢字で「氷の下の魚」と書くタラ科の魚は「スケトウダラ」です。スケトウダラはタラ科に属し、タラ類の中でも特に大きな卵巣を持つことから「助宗鱈 (スケソウダラ)」とも呼ばれています。スケトウダラの卵巣から作られるのがタラコで、白子は明太子として加工されます。

表 2 2 人の作業者による言説分解の結果の例

言説	作業者 1	作業者 2
1	アイスホッケーの試合では、各チームには通常 16 人の選手がいます。	アイスホッケーの試合では、各チームには通常 16 人の選手がいます。
2	アイスホッケーの 16 人の選手の中には、ゴールキーパー（ゴーマン）が 1 人含まれます。	アイスホッケーの試合では、16 人の選手の中にゴールキーパー（ゴーマン）が 1 人含まれます。
3	アイスホッケーの 16 人の選手の中で、15 人はフォワードやディフェンスとしてプレーします。	アイスホッケーの試合では、16 人の選手 中 15 人はフォワードやディフェンスとしてプレーします。
4	アイスホッケーの試合では、試合中に選手交代が頻繁に行われます。	アイスホッケーの試合では、試合中に選手交代が頻繁に行われます。
5	アイスホッケーの試合では、ベンチには多くの選手が控えていることが多いです。	アイスホッケーの試合では、ベンチに多くの選手が控えていることが多いです。

LLM (llm-jp/llm-jp-3-13b-instruct⁽⁴⁾) に生成させることで取得した。データセットに含まれるクイズ問題と生成テキストの例を表 1 に掲載する。

5.1 言説分解が一致したケース

生成テキスト (33) に対する 2 人の作業者（いずれも著者）による言説分解の結果を表 2 に示す。

(33) テキスト：

アイスホッケーの試合では、各チームには通常 16 人の選手がいます。この中には、ゴールキーパー（ゴーマン）が 1 人含まれ、残りの 15 人はフォワードやディフェンスとしてプレーします。試合中に選手交代が頻繁に行われるため、ベンチにはさらに多くの選手が控えていることが多いです。

⁽⁴⁾ <https://huggingface.co/llm-jp/llm-jp-3-13b-instruct>

このうち言説2においては、生成テキスト中の「この中には」の文脈依存性を解消する際の表現に軽微な揺れが見られたものの、両者の作成した言説は概ね一致している。また、言説3においては両者とも「フォワードやディフェンス」を分解しない方針で一致していた。さらに、生成テキスト中の節と節を区切ることににより、言説4と言説5が作成されているという方針も一致していた。

以下の生成テキスト(34)については2人の作業による分解結果が完全に一致し、(35)の結果が得られた。

(34) テキスト：

アガサ・クリスティーが産んだ「灰色の脳細胞」を持つ名探偵は、エルキュール・ポアロです。

- (35) 1. エルキュール・ポアロはアガサ・クリスティーが産みました。
2. エルキュール・ポアロは「灰色の脳細胞」を持ちます。
3. エルキュール・ポアロは名探偵です。

特に言説(35-1)については、「アガサクリスティーがエルキュール・ポアロを産みました。」という言説を作成することも可能だが、両作業者とも「エルキュール・ポアロ」を焦点語として分解を行った結果、言説の表現が一致した。

5.2 言説分解の判断が難しかったケース

言説分解ガイドラインにしたがって言説分解を行っても、判断が一意に定まらない場合があった。以下では代表的な三事例について紹介する。

5.2.1 並列構造の分解方針が異なるケース

言説分解ガイドラインでは、並列構造は原則として分解することとしつつも、一部のケースではテキストの本来の意味を考慮して、並列構造を分解しない言説を採用することもある。4.1.1 節では、テキスト(5)における「4 作品が一つの壮大な物語を形成している」という本来の意味を保持するため、4つの作品名からなる並列構造を分解しない言説(7)を採用することを述べた。

しかし、このような並列構造の分解の可否の判断は作業者に委ねられるため、以下のような生成テキスト(36)の言説分解において、結果に(37)と(38)の揺れが確認された。生成テキスト(36)は下線部の並列構造を含み、分解結果(37)では原則に則ってこれを分解しているのに対し、分解結果(38)では並列構造のまま保持している。

(36) テキスト：

鎌倉時代に成立した「五摂家」とは、近衛家、九条家、一条家、二条家と鷹司(たかつかさ)家のことです。

- (37) 1. 「五摂家」は鎌倉時代に成立しました。
2. 近衛家は「五摂家」です。
3. 九条家は「五摂家」です。
4. 一条家は「五摂家」です。
5. 二条家は「五摂家」です。

6. 鷹司 (たかつかさ) 家は「五摂家」です。

(38) 1. 「五摂家」は鎌倉時代に成立しました。

2. 「五摂家」とは、近衛家、九条家、一条家、二条家と鷹司 (たかつかさ) 家のことです。

分解結果 (37) では、最小粒度への分解を優先し、生成テキスト中の「近衛家、九条家、一条家、二条家と鷹司 (たかつかさ) 家」という並列構造を分解して (37-2) から (37-6) の言説を作成している。仮にテキスト (36) の下線部に誤りが含まれていた場合には、各言説の事実性を確認することで誤りを検出可能であり、妥当な分解であるといえる。

一方、元の生成テキストからは「五摂家」が「近衛家、九条家、一条家、二条家と鷹司 (たかつかさ) 家」を指す語であること、すなわち、「五摂家」の定義が与えられていると読み取れるのに対し、分解 (37) ではそれが失われてしまう。分解結果 (38-2) ではこの点を重視し、生成テキストに含まれる語の定義を過不足なく保持するために並列構造のまとまりを保持している。

このように、並列構造の扱いは生成テキストの意味や文脈に対する作業者の解釈によって異なる場合があり、言説分解結果の揺れにつながった。

5.2.2 焦点語が自明ではないケース

本アノテーションでは焦点語を導入し、焦点語を主語にする語順を基本として言説分解を行うことを定めている。焦点語とは、生成テキストの文やその一部において主題とみなせ、検索における検出力が高いことが期待される表現である (4.2 節参照)。しかし生成テキストによっては焦点語が自明でない場合があり、作業者によって焦点語の定め方が異なることがある。

以下に示す生成テキスト (39) の太字部分に関しては、言説分解の際の焦点語の選択が作業者によって異なったことで、(40) と (41) に示す異なる分解結果が得られた。

(39) テキスト：

アメリカ合衆国の二大政党のうち、**共和党のシンボル**となっている動物は象です。このシンボルは 1874 年に初めて使用され、共和党が「強い政府」と「経済の自由」を象徴するものとして象を選びました。象はその力強さと知性から、共和党の理念や政策を反映するのにふさわしいと考えられました。

(40) 1. 象は共和党のシンボルです。

2. 象は動物です。

(41) 1. 共和党のシンボルは象です。

2. 象は動物です。

(40-1) では「象」を主語としている一方で、(41-1) では「共和党のシンボル」を主語としている。テキストの該当箇所における文構造から「象」が主題であると考えた作業者と、後続文が「このシンボルは」という表現から始まっていることに基づき「共和党のシンボル」が主題であると考えた作業者との間で、焦点語の選択が異なったと考えられる。

上記の分解結果 (40) と (41) の異なり自体は、言説の語順の違いに留まるため、表現上の軽微な差と見なすことができる。しかし、作業者によって焦点語の選択が異なる事例が存在するということは、すなわち 4.2 節で指摘した (17) のような、出現単語の異なる言説が作成される可能性を排除しきれていないことを示唆している。焦点語の選択基準の見直しによる揺れの軽

減は、今後の課題である。

5.2.3 作業者の知識の有無が影響するケース

言説分解作業は特定分野の専門知識を前提としないが、実際には作業者の知識の有無が分解の方針に影響を与える場合がある。

以下の例は、石川さゆりの代表曲である「津軽海峡・冬景色」の歌詞を知っているか否かで、言説分解結果が異なった例である。(43)は「津軽海峡・冬景色」の歌詞を知らない作業者による分解である。生成テキストから「歌詞の中に“青函連絡船の八甲田丸の係留地”が登場する」と解釈し、(43-4)の言説を作成している。

(42) テキスト：

石川さゆりの代表曲で、歌詞の中に登場することから、青函連絡船の八甲田丸の係留地や龍飛崎に歌碑が建てられている曲は「津軽海峡・冬景色」です。

- (43) 1. 「津軽海峡・冬景色」は石川さゆりの代表曲です。
2. 「津軽海峡・冬景色」の歌碑が青函連絡船の八甲田丸の係留地に建てられています。
3. 「津軽海峡・冬景色」の歌碑が龍飛崎に建てられています。
4. 「津軽海峡・冬景色」の歌詞の中には青函連絡船の八甲田丸の係留地が登場します。
5. 「津軽海峡・冬景色」の歌詞の中には龍飛崎が登場します。

しかし、実際には「津軽海峡・冬景色」の歌詞には「青函連絡船」が登場しているのみで「八甲田丸」やその係留地は登場しない。歌詞を知っていた別の作業者は、自身の知識に基づいて生成テキストを解釈した結果、言説4を以下のように作成していた。

(44) 4. 「津軽海峡・冬景色」の歌詞の中には青函連絡船が登場します。

このような異なる言説分解が可能となったのは、生成テキスト(42)の曖昧性に起因する。すなわち、文中の「歌詞の中に登場する」の主語が省略されており、文脈依存性の解消時に補完する表現の候補も構造上は複数存在する。そのため、補完する表現を1つに決定する際に、その選択が作業者の知識によって異なってしまったと考えられる。このように、生成テキストの意味を解釈する際に、作業者の知識を用いて情報が補われる場合には、知識の有無によって分解結果が異なることがある。

6. おわりに

本研究では、LLMの生成テキストの真偽検証システム構築のための日本語言説分解データセットを構築するうえで必要となる言説分解ガイドラインの作成を行った。その際、真偽検証システムにおける言説分解の後段の処理である根拠検索や判定予測において有用な言説が獲得されるように言説分解ルールを設計した。また、分解された言説の揺れを軽減するためのルールも追加した。

作成された言説分解ガイドラインを用いて試験的な言説分解を実施したところ、複雑な構造を持つ生成テキストに対しても複数の作業者間で揺れのない分解を行うことがケースが観測された。その一方で、並列構造の解釈、焦点語の選択、作業者の知識などの差異によって作業者間で揺れが生じた分解も存在した。精緻化で揺れの軽減が期待されるルールについては引き続き改善を行う。また、日本語言説分解データセットの構築のため、大規模データに対する言説

分解の実施や得られた分解結果の定量的な評価についても今後の課題である。

謝 辞

本研究の一部は、令和7年度次世代人工知能技術等研究開発拠点形成事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」、JSPS 科研費（基盤研究 (B) (課題番号: 23H03686, 25K03178), 基盤研究 (C) (課題番号: 24K15066), 株式会社デンソーアイティラボラトリとの共同研究の支援による。ここに記して謝意を表す。

文 献

- Krystal Hu (2023). *ChatGPT sets record for fastest-growing user base - analyst note*. Accessed: 2025-08-07
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaeckermann, Amy Wang, Dev Dash, Jonathan H. Chen, Nigam H. Shah, Sami Lachgar, Philip Andrew Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomašev, Yun Liu, Renee Wong, Christopher Sementuri, S. Sara Mahdavi, Joelle K. Barral, Dale R. Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan (2025). “Toward expert-level medical question answering with large language models.” *Nature Medicine*, 31:3, pp. 943–950.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu (2025). “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions.” *ACM Transactions on Information Systems*, 43, pp. 1–55.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli (2024). *Hallucination is Inevitable: An Innate Limitation of Large Language Models*. arXiv:2401.11817.
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos (2022). “A Survey on Automated Fact-Checking.” *Transactions of the Association for Computational Linguistics (TACL)*, 10, pp. 178–206.
- Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga (2021). “Automated fact-checking: A survey.” *Language and Linguistics Compass*, 15.
- Neema Kotonya, and Francesca Toni (2020). “Explainable Automated Fact-Checking: A Survey.” *Proc. of the COLING 2020*.
- Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett (2023). “WiCE: Real-World Entailment for Claims in Wikipedia.” *Proc. of the EMNLP 2023*.
- Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov (2024).

- “Factcheck-Bench: Fine-Grained Evaluation Benchmark for Automatic Fact-checkers.” *Proc. of the Findings of the EMNLP 2024*.
- Deren Lei, Yaxi Li, Mengya Hu, Mingyu Wang, Vincent Yun, Emily Ching, and Eslam Kamal (2023). *Chain of Natural Language Inference for Reducing Large Language Model Ungrounded Hallucinations*. arXiv:2310.03951.
- Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme (2024). “A Closer Look at Claim Decomposition.” *Proc. of the 3th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*.
- Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du (2025). “Language Ranker: A Metric for Quantifying LLM Performance Across High and Low-Resource Languages.” *Proc. of the AAAI 2025*.
- Chengguang Gan, and Tatsunori Mori (2023). “Sensitivity and Robustness of Large Language Models to Prompt Template in Japanese Text Classification Tasks.” *Proc. of the PACLIC 2023*.
- Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe (2024). “Toward the Evaluation of Large Language Models Considering Score Variance across Instruction Templates.” *Proc. of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal (2018). “FEVER: a Large-scale Dataset for Fact Extraction and VERification.” *Proc. of the NAACL 2018*.
- James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal (2019). “The FEVER2.0 Shared Task.” *Proc. of the Second Workshop on Fact Extraction and VERification (FEVER)*.
- Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal (2021). “The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task.” *Proc. of the Fourth Workshop on Fact Extraction and VERification (FEVER)*.
- Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos (2023). “The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task.” *Proc. of the NeurIPS 2023*.
- Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang (2024). *A Survey on Data Selection for Language Models*. arXiv:2402.16827.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg (2023). “Inference-Time Intervention: Eliciting Truthful Answers from a Language

- Model.” *Proc. of the NeurIPS 2023*.
- Potsawee Manakul, Adian Liusie, and Mark Gales (2023). “SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models.” *Proc. of the EMNLP 2023*.
- Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng (2024). “Self-Alignment for Factuality: Mitigating Hallucinations in LLMs via Self-Evaluation.” *Proc. of the ACL 2024*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2020). “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” *Proc. of the NIPS 2020*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang (2023). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv:2312.10997.
- S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das (2024). *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. arXiv:2401.01313.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2024). “Reducing Hallucination in Structured Outputs via Retrieval-Augmented Generation.” *Proc. of the NAACL 2024*.
- Wan Zhang, and Jing Zhang (2025). “Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review.” *Mathematics*, 13.
- Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho (2025). 「Hallucination - Free? Assessing the Reliability of Leading AI Legal Research Tools」 *Journal of Empirical Legal Studies*, 22, pp. 216–242.
- Yun-Wei Chu, Kai Zhang, Christopher Malon, and Martin Renqiang Min (2025). “Reducing Hallucinations of Medical Multimodal Large Language Models with Visual Retrieval-Augmented Generation .” *Proc. of the AAAI 2025*.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi (2023). “FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation.” *Proc. of the EMNLP 2023*.
- Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster (2023). “PropSegmEnt: A Large-Scale Corpus for Proposition-Level Segmentation and Entailment Recognition.” *Proc. of the Findings of the ACL 2023*.
- Kushan Mitra, Dan Zhang, Sajjadur Rahman, and Estevam Hruschka (2024). “FactLens: Benchmarking Fine-Grained Fact Verification.” *Proc. of the Findings of the ACL 2025*.