

LLM の頑健性と可制御性： ビジネス場面ではどちらを重視すべきか？

馬 春鵬

株式会社リクルート Megagon Labs, Tokyo, Japan

ma.chunpeng@megagon.ai

概要

大規模言語モデルにおけるプロンプト変動が出力に対する影響について、様々な文脈で研究されており、用語は多数存在する。本研究は、既存研究で混在してきた概念を「頑健性」と「可制御性」の二軸から再構造化する。さらに、公開データセットを前提とした従来の分析とは異なり、複雑なタスク構成や追加知識の記述を要するビジネスサービス環境に着目し、両概念の重要度を体系的に評価した。実験の結果、我々が考察したタスクにおいては、先行研究で強調されてきた頑健性よりも、プロンプト意図を確実に反映し必要情報を安定して引き出す可制御性が実運用において本質的であることが明らかとなった。本研究は、ビジネス環境に適したプロンプト設計指針の再考に寄与するとともに、将来の評価指標構築やモデル改善への示唆を提供する。

1 はじめに

大規模言語モデル (Large Language Model, LLM) は、近年、多様な自然言語処理タスクにおいて著しい性能を示し、実世界の応用を急速に拡大している。タスクの指示、出力形式、制約条件などを自然言語で与えるプロンプトは、LLM の挙動を操作するための主要な手段であり、その設計如何によって得られる出力の品質が大きく変動することが知られている [1, 2]。このため、プロンプト設計に関する研究はモデル利用の実践と理論の双方において中心的な位置を占めている。実践については、経験則となるプロンプトエンジニアリングのベストプラクティスが報告されており [3, 4]、理論については、プロンプトの設計が LLM の性能に及ぼす影響 [5, 6] や LLM の性能安定性向上に関する研究 [7, 8] が多数ある。

しかし、様々な背景で行われた多数の先行研究では、必ずしも用語が整理されておらず、sensitivity、

robustness、controllability、brittleness など、互いに関連しながらも焦点の異なる複数の概念が提案されており、研究間で概念の不統一が存在する。これらは総じてプロンプト変動に伴う出力の変化を分析する枠組みとして用いられてきたが、概念間の境界が曖昧であるため、研究成果を体系的に比較・統合することが困難であった。こうした理論的整理の不足は、LLM の挙動理解やプロンプト設計指針の確立にとって重要な障壁となっている。

本稿では、混在されている専門用語を整理・統合し、プロンプト変化と LLM 出力の関係をより構造的に把握するための枠組みとして、頑健性と可制御性の二点を中心概念として位置づける。これによってプロンプト操作方法を再整理し、プロンプトの意味の変化の有無による分類でプロンプトと LLM の挙動に関する論点を明確化する。

概念と論点を明確化した上で、本研究では実際のビジネスサービスにおける LLM の応用に焦点を当てる。先行研究で一般的に用いられてきた公開データセット上の実験設定に異なり、ビジネスサービスにおいては、タスクが複雑であり、LLM に知識を追加する必要がある。そのため、先行研究が LLM に対する分析が必ずしも成立するわけではない。本研究では、ビジネス環境において頑健性と可制御性のいずれがより重要であるのかを体系的に評価し、現実的な利用条件に基づく観点から LLM のプロンプト依存性を再考する。

実験結果から、考察されたタスクにおいては、既存研究で頻繁に指摘されてきた頑健性の不足が必ずしもビジネス環境で最重要課題ではないことが示された。むしろ、プロンプトの意図が確実に反映され、タスクに必要な情報を確実に引き出せる可制御性の方が、ビジネスシステムにおいてはより重大な要素であることが明らかとなった。本研究の知見は、LLM の実運用に向けたプロンプト設計方針の

再検討に資するだけでなく、今後のモデル評価指標の設計やモデル改善の方向性に対しても重要な示唆を与えるものである。

2 LLM の頑健性と可制御性

本章では、頑健性と可制御性の定義を与えた上で、LLM のこれらの性質を測るためにプロンプトを改変する手法を説明する。

2.1 頑健性

プロンプトエンジニアリングする際に、様々な要素を調節しないとイケない。プロンプトの意味を実質的に改変しなくても、ローカルな調節（例：カンマは全角か半角か、改行を入れるかどうか）からグローバルな調節（例：プロンプトの言語は日本語か英語か、箇条書きにするかどうか）まで、プロンプトに変更可能な箇所が多数存在する。直感的に、これらの調節は LLM の出力に大きな影響を及ぼすことが望ましくなく、頑健な LLM の出力は安定すべきだ。よって、以下のように頑健性を定義する。

頑健性 (robustness)

頑健性とは、意味的に等価なプロンプトに対して、LLM が等価な出力を維持する性質を指す。

LLM の頑健性を測定するために、本稿では、意味的に等価なプロンプトを作成し、目標タスクの精度の変化を測る。意味的に等価なプロンプトを作成するために、以下の手法を用いる。

- 【W】 単語¹⁾粒度のノイズを加える
 - 【W.del】 プロンプトにある単語を選び²⁾、削除する
 - 【W.add】 プロンプトにある単語を選び、後ろに同じ単語を加える
 - 【W.err】 プロンプトにある単語を選び、綴りを間違えるようにする
- 【S】 文粒度のノイズを加える
 - 【S.swap】 プロンプトにある文を一つ選び、二つの単語の順番を交換する
 - 【S.para】 プロンプトにある文を一つ選び、言い換えを生成する³⁾
- 【D】 文章粒度の等価変換をする

1) 単語とは、英単語一個、或いは漢字・ひらがな・カタカナのどれか一文字、或いは句読点・改行のどれか一つを指す。

2) 事前に閾値 θ が決まった連続一様分布に従う。

3) 言い換えは GPT 5.1 を用いて生成された。https://openai.com/ja-JP/index/gpt-5-1/

- 【D.shuf】 プロンプトにある知識⁴⁾の項目の順番をシャッフルする
- 【D.form】 プロンプトの形式を変える
- 【D.lang】 プロンプトの言語を変える

2.2 可制御性

言うまでもなく、プロンプトエンジニアリングの目標はタスクの性能を向上させることだ。期待通りの出力を得るために、プロンプトに適切な情報や指示や制約を入れないとイケない。ただし、LLM の出力は必ずしもプロンプトの指示に従うとは限らず、プロンプトに知識を追加しても LLM の出力が変わらないことが屢々観察された。制御性が高い LLM では、出力はプロンプトの変化を忠実に反映すべきだ。よって、以下のように可制御性を定義する。

可制御性 (controllability)

可制御性とは、意味的に差異がある二つのプロンプトに対して、LLM のそれぞれの出力の差異がプロンプトの差異を適切に反映する性質を指す。

LLM の可制御性を測定するために、本稿では、意味的に差異があるプロンプトを作成し、目標タスクの精度の変化を測る。意味的に差異があるプロンプトを作成するために、以下の手法を用いる。

- 【K】 プロンプトにある知識を増減する
 - 【K.del】 プロンプトにある知識の項目を一つ選び、削除する
 - 【K.add】 プロンプトにある知識の項目を一つ選び、同じものをもう一回入れる
- 【K】 間違った知識をプロンプトに入れる
 - 【K.err】 プロンプトにある知識の項目を二つ選び、知識の値を交換する

3 実験

3.1 実験設定

タスク ビジネスサービスを考察するために、本稿では、旅行目的の顧客の問い合わせに対して、顧客の意図を認識するタスクに取り組む。旅行目的の顧客の意図の分類体系があり、LLM の入力顧客の問い合わせであり、分類体系から適切な意図を一

4) 本稿では、知識は項目のリストの形となる。一つの項目はキーと値のように表す。

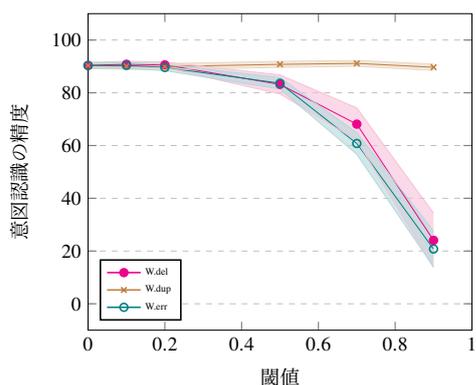


図1 単語粒度のノイズを加えて、Sonnet 3.5 の頑健性を測定した結果。

つ選んで出力する。実際の顧客層や問い合わせの言葉遣いが多様であり、そこから意図を汲み取るのが相当な知識と経験が必要だ。さらに、複数意図の対応や類似する意図の区別などの課題もあり、人間の専門家にとっても難しいケースが屡々ある。

データ 旅行目的の意図の分類体系に、意図の名称が数百個あり、それぞれの意図に対して一つの説明文が付いている。意図の名称は人手で作成されたが、意図の説明文は LLM によって自動作成されたものである。次に、これらの情報に基づいて、意図ごとに 10 個の問い合わせを自動作成した。

LLM の選定 本稿では、API の呼び出しで使える二つの LLM を実験の対象とする：Amazon Bedrock に搭載された Sonnet 3.5⁵⁾ と Microsoft Azure に搭載された GPT 5.1⁶⁾。

評価 出力を評価するために、意図分類の精度を測る。全ての設定において、実験は 5 回行い、精度の平均値と分散を報告する⁷⁾。

3.2 実験結果

我々が考察を行った旅行目的の意図分類タスクにおいては、以下の実験結果が得られた。

3.2.1 LLM の頑健性の結果

2.1 節に説明した通り、様々な粒度のノイズを加え、意味的に等価なプロンプトを作成した。作成されたプロンプトを用いて、LLM の頑健性を測定し

5) <https://aws.amazon.com/jp/blogs/aws/anthropics-claude-3-5-sonnet-model-now-available-in-amazon-bedrock-the-most-intelligent-claude-model-yet/>

6) <https://techcommunity.microsoft.com/blog/azure-ai-foundry-blog/gpt%E2%80%911-in-foundry-a-workhorse-for-reasoning-coding-and-chat/4469067>

7) 本文に報告されたのは Sonnet 3.5 の実験結果である。GPT 5.1 の実験結果は付録 A に記載されている。

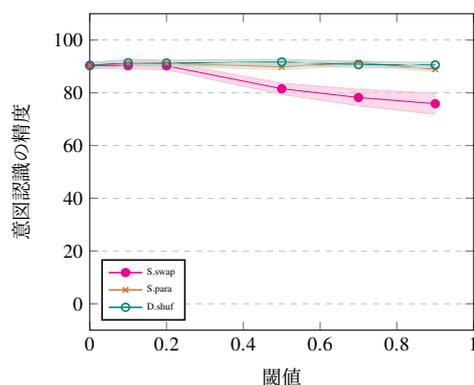


図2 文粒度のノイズや D.shuf ノイズを加えて、Sonnet 3.5 の頑健性を測定した結果。

表1 文章粒度のノイズである D.form と D.lang を加えて、Sonnet 3.5 の頑健性を測定した結果。

設定	意図認識の精度
ベースライン	90.37 ± 1.1
【D.form】簡条書き	90.60 ± 1.3
【D.form】改行無し	90.67 ± 1.5
【D.lang】日本語に翻訳	90.22 ± 0.9
【D.lang】中国語に翻訳	90.00 ± 1.2

た。結果は図 1、2 と表 1 に記載された。これらの結果から、ビジネスサービスのタスクに対して、以下の事実がわかる。

少量なノイズをプロンプトに加えても、LLM は頑健である。 単語粒度のノイズや文粒度のノイズや D.shuf ノイズの場合、閾値（すなわち、ノイズを加える確率）が 0.2 以下であると、意図認識の精度はほぼ変わらない。実際応用を想定すると、単語の綴りの間違い (W.err) や単語の順番の間違い (S.swap) など、偶発的な事象であり、確率は 0.2 を超えることはほぼないため、LLM は頑健だと言える。

同格な文や文章に対して、LLM は頑健である。 S.para や D.shuf の場合、閾値が 0.9 の場合（即ち、大多数の文をその言い換えにする、或いは大多数の知識はナレッジベースの別の場所におかれる場合）、意図認識の精度はほぼ変わらない。さらに、表 1 に示されたように、フォーマットやプロンプトの言語を変えても、顕著な精度変化が見られない。実際応用の場面では、プロンプトの言語や書き方は人によって違うが、本質的な影響はない。

大量なノイズをプロンプトに加えて初めて、LLM の性能が損なわれた。 大量なノイズを加えると意図認識の精度が下がるケースが 3 つある：W.del、W.err、S.swap。ただ、これらのノイズを大量に加えると、新しいプロンプトと元のプロンプトはもはや等価なプロンプトでなくなってしまい、LLM の頑

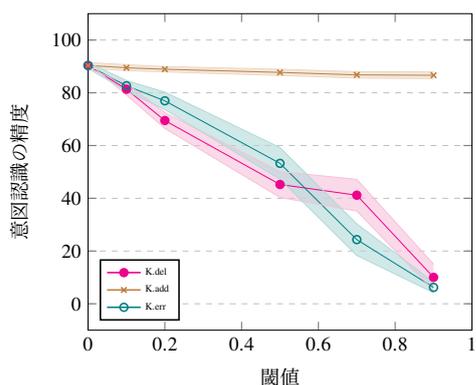


図3 Sonnet 3.5の可制御性を測定した結果。

健性の測定というより、可制御性の測定になる。

3.2.2 LLMの可制御性の結果

LLMの可制御性を測定するために、2.2に説明した手法を用いて、意味的に差異があるプロンプトを作成した。新しいプロンプト(即ち、閾値 > 0)と元のプロンプト(即ち、閾値 $= 0$)を用いて意図認識の精度を測って、LLMの可制御性を測定した。結果は図3に記載された。これらの結果から、ビジネスサービスのタスクに対して、以下の事実がわかる。

プロンプトにある知識を調節することによって、LLMを制御することができる。 閾値が0.1のような小さい値でも、意図認識の精度が大幅に下がる。これは提供された知識が不足(K.del)や間違った知識が含まれている(K.err)と意味する。よって、プロンプトに知識を加えたり修正したりすることによって、LLMの精度を向上させることができ、制御可能である。

プロンプトエンジニアリングより、高品質な知識の準備は性能向上の近道である。 今の時代では、多様タスクの公開データに高い性能を獲得したと主張されたLLMは散見されるが、実際のビジネスサービスのタスクに対して、専門的な知識がないと(例えば、K.delの閾値が高い場合)、高い精度は得られない。公開LLMを実製品に適用する際に、プロンプトエンジニアリングに時間を費やすよりも、専門的な知識を準備した方が、(図3の閾値を下げることによって)性能向上に貢献するであろう。ただし、準備された知識は大量(K.delの曲線を参照)で正確(K.errの曲線を参照)でなければならない。

4 先行研究

4.1 LLMの頑健性に関する研究

LLMの頑健性は様々な文脈において研究されており、等価なプロンプトを作成する様々な手法が別々で考察された: 句読点の変化[1, 9]、フォーマットの変化[2]、単語のノイズ[10]、言い換え[11]、知識の並び替え[5]、few-shot[12]。敏感性(sensitivity)や脆弱性(brittleness)という用語を使う研究もあり、可制御性に混同する研究もある。

既存研究では、これらの変化に対してLLMの出力が安定ではないことが報告されたが、ビジネス現場の感覚に食い違いがある。本研究ではビジネスサービスの観点から、等価なプロンプトの作成手法を統合し、LLMの頑健性について考察を行った。

4.2 LLMの可制御性に関する研究

プロンプトにある情報がLLMに無視される(即ち、可制御性が低い)課題はかねてより知らされている。事前学習によって、大量な知識がLLMに織り込まれ[13, 14]、そのためプロンプトにある情報が無視されてしまう[15, 16]。

LLMの可制御性を向上させるために、プロンプト設計の改良[15]や学習時の工夫[17, 18]や推論時の工夫[19, 20]など、多様な手法が提案された。本研究ではビジネスサービスにおいてもLLMの可制御性の課題が存在することを示し、これからLLMの可制御性の向上に関する研究の重要性がわかった。

5 おわりに

本稿では、プロンプトの変化とLLM出力の関係というテーマにまつわる概念群を頑健性と可制御性の二点を中心概念として整理し、正確な定義を提唱した。その上で、ビジネスサービスにおけるLLMの応用に焦点を当て、ビジネス場面によく使われているLLMの頑健性と可制御性を測定した。我々が考察した旅行目的の意図分類タスクにおいては、既存研究で頻繁に指摘された頑健性不足の課題はビジネスサービスのタスクにおいては最重要課題ではなく、可制御性の方が重要であることが明らかとなった。よって、ビジネス環境では、プロンプトエンジニアリングより、大量な高品質な知識の準備は性能向上の近道であることがわかった。

参考文献

- [1] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In **The Twelfth International Conference on Learning Representations**, 2024. <https://openreview.net/forum?id=RIu5LyNXjT>.
- [2] Anton Voronov, Lena Wolf, and Max Ryabinin. Mind your format: Towards consistent evaluation of in-context learning improvements. In **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 6287–6310, 2024. <https://aclanthology.org/2024.findings-acl.375/>.
- [3] Louie Giray. Prompt engineering with chatgpt: a guide for academic writers. **Annals of biomedical engineering**, Vol. 51, No. 12, pp. 2629–2633, 2023.
- [4] Jiaqi Wang, Enze Shi, Sigang Yu, Zihao Wu, Huawei Hu, Chong Ma, Haixing Dai, Qiushi Yang, Yanqing Kang, Jinru Wu, et al. Prompt engineering for healthcare: Methodologies and applications. **Meta-Radiology**, p. 100190, 2025.
- [5] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In **Findings of the Association for Computational Linguistics: NAACL 2024**, pp. 2006–2017, 2024. <https://aclanthology.org/2024.findings-naacl.130/>.
- [6] Sheng Lu, Hendrik Schuff, and Iryna Gurevych. How are prompts different in terms of sensitivity? In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 5833–5856, 2024. <https://aclanthology.org/2024.naacl-long.325/>.
- [7] Chenxing Wei, Mingwen Ou, Ying He, Yao Shu, and Fei Yu. PAFT: Prompt-agnostic fine-tuning. In **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pp. 694–717, 2025. <https://aclanthology.org/2025.emnlp-main.37/>.
- [8] Lilian Ngweta, Kiran Kate, Jason Tsay, and Yara Rizk. Towards LLMs robustness to changes in prompt format styles. In **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)**, pp. 529–537, 2025. <https://aclanthology.org/2025.naacl-srw.51/>.
- [9] Mikhail Seleznyov, Mikhail Chaichuk, Gleb Ershov, Alexander Panchenko, Elena Tutubalina, and Oleg Somov. When punctuation matters: A large-scale comparison of prompt robustness methods for LLMs. In **Findings of the Association for Computational Linguistics: EMNLP 2025**, pp. 20370–20385, 2025.
- [10] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt perturbation consistency learning for robust language models. In **Findings of the Association for Computational Linguistics: EACL 2024**, pp. 1357–1370, 2024.
- [11] Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. **Transactions of the Association for Computational Linguistics**, Vol. 12, pp. 933–949, 2024.
- [12] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8086–8098, 2022.
- [13] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 2463–2473, 2019.
- [14] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 5484–5495, 2021.
- [15] Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 14544–14556, 2023.
- [16] Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8420–8436, 2024.
- [17] Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 15173–15184, 2023.
- [18] Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. Large language models with controllable working memory. In **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 1774–1793, 2023.
- [19] Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. Activation scaling for steering and interpreting language models. In **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 8189–8200, 2024.
- [20] Julian Minder, Kevin Du, Niklas Stoehr, Giovanni Monea, Chris Wendler, Robert West, and Ryan Cotterell. Controllable context sensitivity and the knob behind it. In **The Thirteenth International Conference on Learning Representations**, 2025.

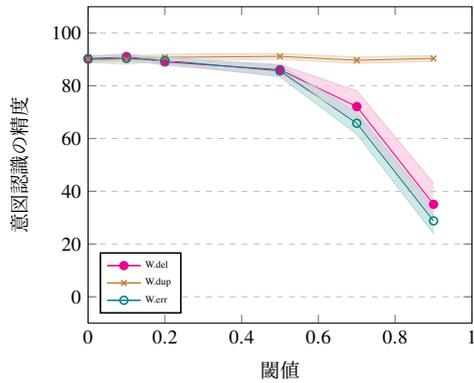


図4 単語粒度のノイズを加えて、GPT 5.1 の頑健性を測定した結果。

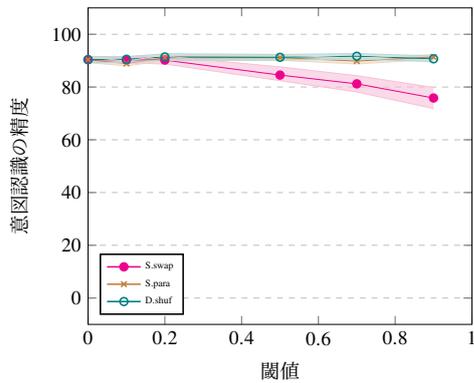


図5 文粒度のノイズや D.shuf ノイズを加えて、GPT 5.1 の頑健性を測定した結果。

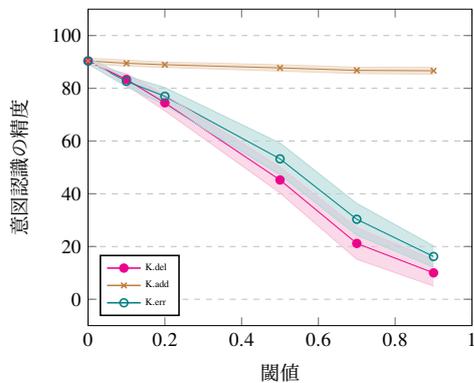


図6 GPT の可制御性を測定した結果。

A GPT 5.1 の実験結果

GPT 5.1 に対して、頑健性の測定結果は図 4、5 と表 2 に記載され、可制御性の測定結果は図 6 に記載された。

数値は Sonnet 3.5 と若干異なるが、傾向は同様であり、本文に書かれた結論は GPT 5.1 に対しても成立する。

B 実験データの例

実験データの作成方法は本文に書かれた通りである。人手で作った旅行目的の意図の名称と自動生成された意図の説明文の例は表 3 に記載され、これらの情報に基づいて自動作成された問い合わせの例は表 4 に記載された。

表 2 文章粒度のノイズである D.form と D.lang を加えて、GPT 5.1 の頑健性を測定した結果。

設定	意図認識の精度
ベースライン	90.05 ± 1.0
【D.form】簡条書き	91.32 ± 1.3
【D.form】改行無し	90.14 ± 1.0
【D.lang】日本語に翻訳	91.03 ± 1.2
【D.lang】中国語に翻訳	90.63 ± 0.7

表 3 旅行目的の意図の分類体系 (例)。名称は人手で作成されたが、説明文は LLM によって自動作成。

【人数を変更したい】 This intent involves customers wanting to change the number of people included in their reservation. Inquiries often arise when there are changes in the group size, such as someone canceling or additional guests joining. Examples include requests to reduce the number of guests due to cancellations, increase the number of guests, or adjust room allocations to accommodate the new group size. Customers may also seek information on how these changes affect the pricing and whether the booking system allows for such modifications. This intent focuses on ensuring the reservation accurately reflects the current number of guests.

【宿・プランを検索したい】 This intent is for users who want to search for accommodations or specific plans and need assistance with the search process. Inquiries under this intent often involve users who are looking for specific features or conditions in their accommodations, such as pet-friendly hotels, rooms with independent bathrooms, or rooms with private hot springs. For example, a user might request the addition of a search filter for rooms with separate baths and toilets. Another user might find the current search options insufficient and request more detailed filters to better match their needs. This intent captures the need for guidance on how to effectively search for and find accommodations that meet specific criteria.

表 4 自動作成された架空の問い合わせの例

【人数を変更したい】 お世話になっております。現在予約している宿泊について、人数の変更をお願いしたご連絡いたしました。当初の予約内容から人数を変更したいのですが、オンライン上で手続きしようとしたところ、システムエラーが発生して変更が完了できませんでした。そのため、直接ご確認いただければ幸いです。つきましては、人数変更に伴う料金の差額や、プラン変更の必要性があれば併せてご教示ください。また、可能であれば最適な部屋割りや料金になるようご提案いただくと助かります。お手数をおかけしますが、どうぞよろしくお願いいたします。

【宿・プランを検索したい】 いつもお世話になっております。旅行を計画しているにあたり、貴館の宿泊プランについて詳しく確認したく、ご連絡を差し上げました。現在、複数の条件のもとに宿泊先を探しております。特に、客室の設備（独立したバス・トイレ、露天風呂付き客室、貸切風呂の有無など）、ベッド同伴の可否、コネクティングルームの有無、複数名での部屋割りに対応できるかといった点を重視しております。また、これらの条件に合致するプランを効率的に探せる方法や、プランの詳細がどのように検索画面で表示されるかについても教えていただければ幸いです。可能でしたら、上記条件を踏まえたおすすめプランの有無や、最適な検索方法・手順についてご教示いただけますと助かります。また、検索結果が最安値順で正確に表示されるかどうか、特別プランなどの表示方法についても併せてお伺いできれば幸いです。お忙しいところ恐れ入りますが、ご確認のほどよろしくお願い申し上げます。