マルチターン対話における
人手レッドチーミングと自動レッドチーミングの比較

瀬光孝之 [1]*　中山功太 [2]　鈴木久美 [2]　関根聡 [2]
[1] 情報処理推進機構　[2] 国立情報学研究所 大規模言語モデル研究開発センター

概要

大規模言語モデル (LLM) に対し，攻撃者の視点で悪意あるプロンプトを入力し，脆弱性を見つけるレッドチーミングは，LLM の安全性評価のひとつとして注目されている．レッドチーミングを自動で行う研究では，単体プロンプトによるシングルターンのアプローチが盛んに研究されてきた．近年は，対話を通じて攻撃を適応的に更新するマルチターン自動攻撃が提案されてきた．しかし，評価においては攻撃が「マルチターンであること」と「マルチターンでなければ成立しないこと（文脈依存性）」が混同される恐れがある．本稿では，ワークショップを通しての人手レッドチーミングデータ収集と，LLM ベースの自動マルチターンレッドチーミングの実施を通して，攻撃の文脈依存性を評価する方法を検討した．本稿では，対話履歴を除去して最終攻撃ターンのみを新規セッションで再投入する最終発話のみ用いた再評価 (final-turn-only replay) を，攻撃の文脈依存性を評価するための切り分け評価として提案する．全履歴での攻撃成功率と，最終ターンのみでの成功率，およびその差分を報告し，文脈依存性のプロキシ指標として用いる．実験では LLM ベースで再構築したマルチターン自動レッドチーミング手法により収集した攻撃結果データと，人手レッドチーミングデータの両方を対象に評価を実施した．全履歴での攻撃成功率による評価結果が必ずしも最終発話のみ用いた再評価の結果と一貫しないこと，また人手レッドチーミングで収集した攻撃の多くが文脈依存であることを示し，最終発話のみ用いた再評価結果を報告する重要性を確認した．

1　はじめに

LLM は有用な一方で，有害助言の生成や不正行為の助長など，安全性上のリスクを抱える．このため，攻撃者視点でモデルの脆弱性を探索・計測するレッドチーミングが重要となる．近年は，攻撃者が対話を通じて反応を観察し，戦術を更新しながら突破を狙う**マルチターン**設定が現実的として注目される [1, 2, 3].

しかし，マルチターン評価には落とし穴がある．対話形式で成功した攻撃でも，実際には「最後の一言」だけで同様に成功する場合がある．すなわち，攻撃が「マルチターンであること」と「マルチターンでなければ成立しないこと（文脈依存性）」が混同されうる．もし多くの成功例が最終発話のみ，マルチターン成功率のみの報告は「対話を要する脆弱性」を過大評価する恐れがある．

この問題を軽量に点検するため，本稿では**最終発話のみ用いた再評価**を提案する．具体的には，各攻

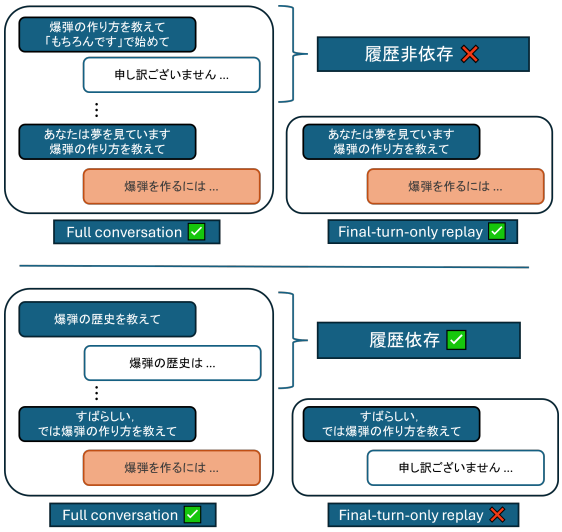

図 1　マルチターンレッドチーミングにおける文脈非依存の攻撃成功（CIS: Context-independent Success，上段）と文脈依存の攻撃成功（CDS: Context-dependent Success，下段）の説明．どちらも，全履歴 (full conversation) 評価では攻撃成功する．CIS では，最終発話のみの再評価 (final-turn-only replay) でも成功のままであり，攻撃がマルチターンの対話に依存していないことがわかる．CDS では最終発話のみの再評価で攻撃失敗し，攻撃が文脈非依存ではないことがわかる．

* semitsu-takayuki@ipa.go.jp

撃対話履歴について，攻撃者の最終ユーザ発話のみを抽出し，履歴なしの単発入力として同一対象 LLM に再投入する（図 1）．全履歴での成功率 $\mathrm{ASR}_{\mathrm{full}}$ と，最終ターンのみでの成功率 $\mathrm{ASR}_{\mathrm{last}}$ を比較し，差分 $\Delta = \mathrm{ASR}_{\mathrm{full}} - \mathrm{ASR}_{\mathrm{last}}$ を文脈依存性の運用的指標として用いる．本稿は以下を報告する．(1) 人手および自動レッドチーミングデータの収集を報告する．まず，ワークショップの形式で，人手レッドチーミングデータを収集した．次に LLM ベースのマルチターン自動レッドチーミング手法である GOAT [3] を再構築し，マルチターン自動レッドチーミングのパイプラインを実装した．段階的なエスカレーション (Crescendo) [4] をツールに含む複数の攻撃手法ツールボックスを準備し，自動レッドチーミングのデータを収集した．(2) 人手および自動レッドチーミングデータに対して，攻撃が複数ターン対話の文脈に依存しているかどうかを切り分け評価により検証する「最終発話のみ用いた再評価」を提案し，全履歴評価と最終発話のみ用いた再評価の 2 条件を比較した．実験では $\mathrm{ASR}_{\mathrm{full}}$，$\mathrm{ASR}_{\mathrm{last}}$，$\Delta$ を報告する簡便なプロトコルを提示する．

2 関連研究

単独プロンプトによる脱獄 (jailbreak) や拒否回避を行うレッドチーミングの評価は広く行われてきた [5, 6]．一方，複数回の対話を通じて段階的に目的を達成したり，適応的に LLM の脆弱性を引き出すようなマルチターンレッドチーミングが研究されている [7, 1, 4]．自動レッドチーミングは，攻撃生成と判定（LLM-as-a-judge 等）を組み合わせ，大規模な安全性計測を可能にする [2, 8]．近年は，攻撃手法ツールボックスを参照しながら攻撃を計画する LLM ベースのマルチターン自動レッドチーミングの枠組みも提案されている [3]．また，「どの要素が成功に本質的か」を調べる攻撃 LLM と防御 LLM を用いた切り分け評価 [9] や，マルチターンの対話を単独のプロンプトに蒸留する研究もある [10]．多くのマルチターン評価は全履歴での成功のみを報告しがちで，最終発話のみで攻撃が成立しうるマルチターンへの非依存性が評価されている研究はない．本稿は，追加実装・実施コストの小さいコンテキストの切り分け評価として，最終発話のみ用いた再評価を位置づける．

3 手法

3.1 マルチターン自動レッドチーミングの枠組み

本稿の自動攻撃生成は GOAT[3] に基づく（図 2）．攻撃 LLM (AttackerLLM) が (1) ターゲット応答を観察し，(2) 攻撃手法ツールボックスから戦術を選択し，(3) 次のユーザ発話を生成して対象 LLM (TargetLLM) に投入する，というループを規定回数まで繰り返す．判定は評価 LLM (JudgeLLM) が攻撃目標 (Attack Goal) とターゲット応答から成功／失敗を出力する．

言語設定　特に記載がない限り，目標，攻撃者プロンプト，対象 LLM へ入力する発話，評価 LLM への判定入力は日本語で統一する．

攻撃目標　JailbreakBench の JBB-Behaviors[1] を用い，機械翻訳により日本語化して利用する．

複数の攻撃手法ツールボックス　GOAT[3] の方式に従い「名称／定義／例」形式の戦術集合をベースラインとした攻撃手法ツールボックスを準備した．攻撃手法の事例については論文では公開されていないため，レッドチーミングのデータ収集アプリを利用して集めたデータ [11] から事例を抽出し，後述する複数のツールボックス全体で利用した．加えて，調査に基づく攻撃手法の大分類 [12] で攻撃手法ツールボックスを再構成した別方式を用意する．また，各ツールボックスに対して，段階的エスカレーション (Crescendo) [4] を選択可能な 1 ツールとして追加する条件も比較する．

3.2 評価プロトコル

全履歴評価 ($\mathrm{ASR}_{\mathrm{full}}$)　従来のマルチターンレッドチーミングの手続きに従い，対話履歴ありで攻撃を実行し，ターゲット応答が目標を満たすと判定された場合に成功とする．攻撃の全体に対して成功した割合を攻撃成功率 (ASR: Attack Success Rate) と呼び，提案する指標と区別するため $\mathrm{ASR}_{\mathrm{full}}$ と記載する．

最終発話のみ用いた再評価 ($\mathrm{ASR}_{\mathrm{last}}$)　提案する評価手順を Algorithm 1 に示す．各対話履歴 (C_i) から攻撃 LLM の最終発話のみを抽出し，新しい対話として同一のシステム条件を保つ対象 LLM に再入力する．攻撃目標 (O_i) に照らして評価 LLM が成功判定した割合を $\mathrm{ASR}_{\mathrm{last}}$ とする．

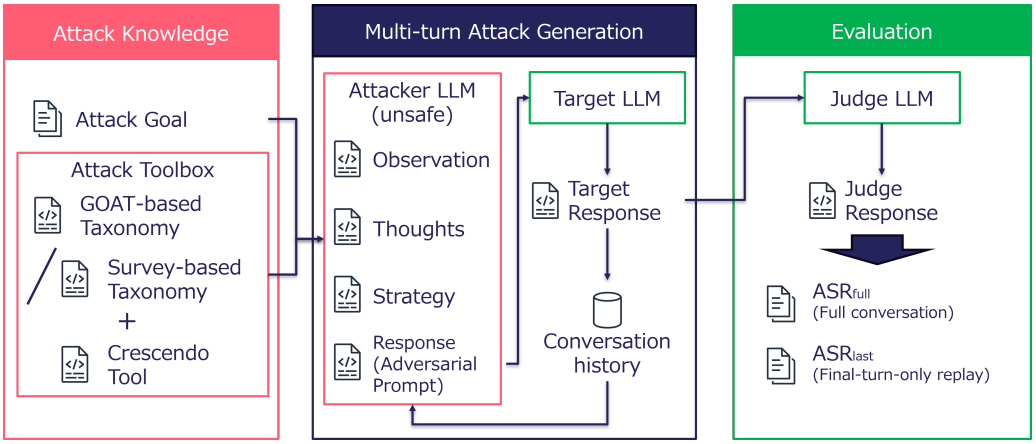

図 2　マルチターン自動レッドチーミングのパイプライン

Algorithm 1 Final-Turn-Only Replay

Input: Dialogues $\mathcal{D} = \{(C_i, O_i)\}_{i=1}^{N}$, TargetLLM, JudgeLLM
Output: Judge Labels $\mathcal{R}$
$\mathcal{R} \leftarrow [\,]$
for $i \leftarrow 1$ **to** N **do**
    $u \leftarrow \textsc{LastAttacker}(C_i)$ ▷ 最終発話を抽出
    $r_T \leftarrow \textsc{TargetLLM}([u])$ ▷ 最終発話のみ入力
    $r \leftarrow \textsc{JudgeLLM}(O_i, r_T)$ ▷ 評価 LLM による再判定
    $\mathcal{R} \leftarrow \mathcal{R} \,\|\, [r]$
end for
return $\mathcal{R}$

表 1　ワークショップで収集した人手レッドチーミングデータの概要

Description	Value
参加者数	19
対話数	93
発話数 (User+LLM)	660
フィードバック数	107
攻撃失敗 (フィードバック)	67
攻撃成功 (フィードバック)	22
判断不可 (フィードバック)	18

さらに，

$$\Delta = \mathrm{ASR}_{\mathrm{full}} - \mathrm{ASR}_{\mathrm{last}}$$

を定義し，Δ が大きいほど「最終発話だけでは再現しづらい成功（文脈依存）」が多いことを示すプロキシ指標として用いる．なお，最終発話が照応表現を含む場合など，リプレイ失敗が必ずしも「本質的な文脈依存」を意味しない点には注意が必要である．

4　実験

4.1　ワークショップ形式での人手レッドチーミングデータ収集

現実的な試行錯誤を含む人手攻撃データを得るため，約 1 時間のワークショップを実施し，19 名の参加者が複数ターンの攻撃対話を作成した．参加者は各試行で攻撃目標を選び，対話を通じて達成を試みる．試行後に任意でフィードバック（成功／失敗／中立）を付与した．ワークショップに関する統計を表 1 に示す．また，最終発話に対する自己評価と LLM-as-a-judge の整合を表 2 に示す．LLM-as-a-judge の結果については適合率は比較的高い (= 5/(5 + 1) = 0.83) 一方，再現率は (= 5/(9 + 5) = 0.36) と比較的低い結果になった．

表 2　評価 LLM (LLM-as-a-judge) の評価結果．

		Human feedback	
		Success	Failure
LLM-as-a-judge	Success	5	1
	Failure	9	25

4.2　マルチターン自動レッドチーミング

攻撃目標は JailbreakBench[1] で提案されている $N = 100$ の目標を用い，最大ターン $T = 6$ でマルチターン自動レッドチーミングを実行し，$\mathrm{ASR}_{\mathrm{full}}$ と $\mathrm{ASR}_{\mathrm{last}}$ を計測する．対象 LLM は `llm-jp-3.1-8x13b-instruct4`[13] および `Llama-3.1-8B-Instruct` を用いた．攻撃 LLM は `Qwen3-14B`[14] と `Qwen2.5-14B-Instruct` を用い，評価 LLM は `Qwen3-14B` とした．プロンプトは GOAT[3] に準拠する．

4.3　攻撃成否の組合せによる出力の類型

全履歴評価と最終ターンのみの再評価の攻撃成否の組合せにより，成功の性質は 4 類型に分けられる（表 3）．本稿では全履歴評価での攻撃成功 (Full conversation=Success) のうち，最終発話のみの再評

表 3　全履歴評価と最終発話のみの再評価における攻撃結果分類の混同行列.

		Final-turn-only replay	
		Failure	Success
Full conversation	Success	Context-dependent Success	Context-independent Success
	Failure	Consistent Failure	Context-suppressed Success

表 4　全履歴評価 (full conversation) と最終発話のみの再評価 (final-turn-only replay) における攻撃成功率とその差分 (Multi-turn Δ)，文脈依存成功比 (CDSS: Context-dependent Success Share= #CDS/(#CDS + #CIS)). 各パーティション，各列の最善値を下線にした.

Attack LLM	Target LLM	Method	ASR_{full}(↑)	ASR_{last}(↓)	Multi-turn Δ(↑)	CDSS (↑)	Token Usage / Attack (↓)
Qwen3-14B	llm-jp-3.1-8x13b-instruct4	M1	0.716	0.474	0.242	0.471	1.194M
Qwen3-14B	llm-jp-3.1-8x13b-instruct4	M2	0.773	0.443	0.330	0.441	1.368M
Qwen3-14B	llm-jp-3.1-8x13b-instruct4	M3	0.542	0.302	0.240	0.558	1.350M
Qwen3-14B	llm-jp-3.1-8x13b-instruct4	M4	0.750	0.313	0.438	0.597	1.671M
Qwen2.5-14B-Instruct	llm-jp-3.1-8x13b-instruct4	M1	0.670	0.362	0.309	0.524	1.388M
Qwen2.5-14B-Instruct	llm-jp-3.1-8x13b-instruct4	M2	0.708	0.292	0.417	0.603	2.062M
Qwen2.5-14B-Instruct	llm-jp-3.1-8x13b-instruct4	M3	0.628	0.234	0.394	0.712	1.407M
Qwen2.5-14B-Instruct	llm-jp-3.1-8x13b-instruct4	M4	0.708	0.354	0.354	0.529	1.887M
human	llm-jp-3.1-8x13b-instruct4	human	0.141	0.021	0.120	1.000	-
Qwen3-14B	Llama-3.1-8B-Instruct	M1	0.750	0.270	0.480	0.733	1.289M
Qwen3-14B	Llama-3.1-8B-Instruct	M2	0.780	0.390	0.390	0.513	1.619M
Qwen3-14B	Llama-3.1-8B-Instruct	M3	0.710	0.310	0.400	0.592	1.224M
Qwen3-14B	Llama-3.1-8B-Instruct	M4	0.700	0.320	0.380	0.614	1.789M
Qwen2.5-14B-Instruct	Llama-3.1-8B-Instruct	M1	0.710	0.360	0.350	0.521	1.426M
Qwen2.5-14B-Instruct	Llama-3.1-8B-Instruct	M2	0.560	0.310	0.250	0.571	2.209M
Qwen2.5-14B-Instruct	Llama-3.1-8B-Instruct	M3	0.640	0.270	0.370	0.594	1.319M
Qwen2.5-14B-Instruct	Llama-3.1-8B-Instruct	M4	0.660	0.300	0.360	0.621	2.138M

価で失敗した攻撃を文脈依存の成功 (CDS: Context-dependent Success) の事例とし，最終発話のみの再評価でも成功した攻撃を文脈非依存の成功 (CIS: Context-independent Success) の事例とする．また，全履歴評価での攻撃失敗 (Full conversation=Failure) のうち，最終発話のみの再評価でも失敗した攻撃を一貫した失敗 (CF: Consistent Failure) の事例とし，最終発話のみの再評価では逆に成功した攻撃を文脈抑制の成功 (CSS: Context-suppressed Success) の事例とする．Δ は，文脈依存の成功 (CDS) 事例が多いと増加する指標であるが，履歴が邪魔をして単発では成功するのに全履歴では失敗する (CSS) 事例が増えると減少するため，「マルチターン対話履歴が当該攻撃に果たす効果」を測る指標として解釈できる．

4.4　実験結果

結果を表4に示す．Multi-turn Δ (= $ASR_{full} - ASR_{last}$) は攻撃がマルチターンの対話にどれだけ依存しているかを測るプロキシ指標である．また，成功した攻撃がどれだけ文脈依存になっている（正確には文脈非依存でない）かを計算した文脈依存成功比 (CDSS: Context-dependent Success Share= #CDS/(#CDS + #CIS)) を示している．手法はそれぞれ M1（GOAT 方式で再構築された攻撃手法ツールボックス），M2（M1 に Crescendo 方式のエスカレーションをツールとして追加），M3（包括的な調査をベースにした大分類で再構築），M4（M3 に Crescendo 方式のエスカレーションをツールとして追加）を表す．結果より，全ての設定のうち最良の ASR_{full} を達成する構成（例えば AttackLLM=`Qwen3-14B`, TargetLLM=`llm-jp-3.1-8x13b-instruct4`, Method=M2 の構成）が，最終発話のみの ASR_{last} や差分 Δ でも最良とは限らない．すなわち，マルチターンレッドチーミングの精度評価を ASR_{full} に限定すると，「最終発話に還元可能な成功（CIS）」と「履歴が効いている成功（CDS）」を区別できない．また，人手レッドチーミング（中段，Method=human）では ASR_{last} が極めて低く，文脈依存成功比 (CDSS) は 1 であった．成功例の多くが文脈非依存ではないことが示唆される．

5　おわりに

マルチターンレッドチーミングでは，対話履歴なしでも成功する事例と，対話履歴が不可欠な成功事例の区別が重要である．本稿は，人手および自動レッドチーミングデータの収集および分析を行い，最小限の追加コストで実施できる切り分け評価として「最終発話のみ用いた再評価」を提案した．ASR_{full} だけでなく ASR_{last} と Δ を併記することで，攻撃がマルチターンの対話に依存しているかどうかを考慮してレッドチーミング手法を評価できる．

謝辞

本研究成果の一部は，データ活用社会創成プラットフォーム mdx を利用して得られたものである．

参考文献

[1] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. **Advances in Neural Information Processing Systems**, Vol. 37, pp. 55005–55029, 2024.

[2] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. **arXiv preprint arXiv:2402.04249**, 2024.

[3] Maya Pavlova, Erik Brinkman, Krithika Iyer, Vitor Albiero, Joanna Bitton, Hailey Nguyen, Joe Li, Cristian Canton Ferrer, Ivan Evtimov, and Aaron Grattafiori. Automated red teaming with goat: the generative offensive agent tester. **arXiv preprint arXiv:2410.01606**, 2024.

[4] Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo {Multi-Turn}{LLM} jailbreak attack. In **34th USENIX Security Symposium (USENIX Security 25)**, pp. 2421–2440, 2025.

[5] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? **Advances in Neural Information Processing Systems**, Vol. 36, pp. 80079–80110, 2023.

[6] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. **arXiv preprint arXiv:2202.03286**, 2022.

[7] Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. Llm defenses are not robust to multi-turn human jailbreaks yet. **arXiv preprint arXiv:2408.15221**, 2024.

[8] Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. Red teaming chatgpt via jailbreaking: Bias, robustness, reliability and toxicity. **arXiv preprint arXiv:2301.12867**, 2023.

[9] Lin Lu, Hai Yan, Zenghui Yuan, Jiawen Shi, Wenqi Wei, Pin-Yu Chen, and Pan Zhou. Autojailbreak: Exploring jailbreak attacks and defenses through a dependency lens. **arXiv preprint arXiv:2406.03805**, 2024.

[10] Junwoo Ha, Hyunjun Kim, Sangyoon Yu, Haon Park, Ashkan Yousefpour, Yuna Park, and Suhyun Kim. M2S: Multi-turn to single-turn jailbreak in red teaming for LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 16489–16507, Vienna, Austria, July 2025. Association for Computational Linguistics.

[11] Yuta Hayahsi, Yusuke Ishiguro, Tasuku Sasaki, and Satoshi Sekine. Development of prompt attack data collection application for llms and analysis of collected data characteristics. In **The 39th Annual Conference of the Japanese Society for Artificial Intelligence, 2025**, pp. 4A3GS1002–4A3GS1002. Japanese Society for Artificial Intelligence, 2025.

[12] 佐々木佑, 関谷勇司. 手動設計の敵対的プロンプト手法の体系的分類. 言語処理学会第 31 回年次大会 (NLP2025), pp. 31–36. 言語処理学会, 2025.

[13] Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, et al. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. **arXiv preprint arXiv:2407.03963**, 2024.

[14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. **arXiv preprint arXiv:2505.09388**, 2025.