進化戦略に基づくLLM出力の自動修正による有害度低減

芝原 俊樹[1,a)]　松林 勝[1]　岩花 一輝[1]　小山 卓麻[1]　大湊 健一郎[1]

概要：大規模言語モデル（LLM）を安全に活用するために，倫理的に不適切な表現を検出して応答を拒否するガードレールが広く導入されている．これにより，差別的・暴力的発言など明らかに有害と判断される出力を遮断できる．一方で，判定者の主観によって問題の有無が分かれる「中程度の有害度」を含む表現まで過度に検知すると，ユーザーへの回答拒否が増加し利便性が低下してしまう．企業や自治体では，こうした中程度の有害度を含む応答も許容できないため，適切に排除しつつ過剰な拒否を回避する仕組みが求められている．本研究では，中程度の有害度をもつ出力を安全な表現に自動修正する進化戦略に基づく手法を提案する．具体的には，まず修正前の出力から複数のパラフレーズを生成し，それぞれの有害度を評価したうえで最も有害度が低いパラフレーズを選択する．次に，選択されたパラフレーズを参照しつつ新たな出力を再生成する．この手順を繰り返すことで，多くの人で安全と判断される出力を得ることができる．8B–32B の 3 種の LLM と 2 つのデータセットで評価した結果，既存手法と比較して最大 16.3% 修正成功率が改善した．さらに，ablation study により低有害度パラフレーズの選択とパラフレーズを参照した再生成の双方が安定して修正を成功させるために不可欠であることを確認した．

Reducing Toxicity in LLM Outputs through Evolution-Strategy-Based Automatic Correction

Toshiki Shibahara[1,a)]　Masaru Matsubayashi[1]　Kazuki Iwahana[1]　Takuma Koyama[1]
Kenichiro Ominato[1]

Abstract: To ensure the safe deployment of large language models (LLMs), guardrails that detect ethically inappropriate expressions and block responses are widely used. These guardrails can effectively prevent clearly harmful outputs, such as discriminatory or violent statements. However, when detection criteria are overly strict and include expressions with moderate toxicity, where opinions on their acceptability may differ among evaluators, the frequency of refusals increases, reducing usability. In corporate and public institution applications, outputs containing even moderate toxicity are often unacceptable, making it essential to remove them while avoiding excessive refusals. In this study, we propose an evolution-strategy-based method for automatically correcting LLM outputs with moderate toxicity into safe expressions. The method first generates multiple paraphrases of the original output, evaluates the toxicity of each, and selects the paraphrase with the lowest toxicity. A new output is then regenerated with reference to the selected paraphrase. By repeating this process, the proposed method can generate outputs that are widely regarded as safe. We evaluated the proposed method on three LLMs with parameter sizes from 8B to 32B and two datasets. Compared with existing methods, our approach achieved up to a 16.3% improvement in correction success rate. An ablation study further confirmed that both selecting low-toxicity paraphrases and regenerating outputs with reference to paraphrases are essential for consistently achieving successful revisions.

1. はじめに

近年，LLM（Large Language Model）の社会実装が進む中，倫理・法務上のリスクを低減するためにガードレールが広く用いられている．一般的なガードレールは，LLM の出力を監視し，規約違反や倫理的に問題のある内容を検知した場合に「応答できません」といった固定的な拒否応答に置き換える．しかし，実運用では表現の是非について人

[1] NTT 社会情報研究所
NTT Social Informatics Laboratories
[a)] toshiki.shibahara@ntt.com

© 2025 Information Processing Society of Japan

This paper is work in progress and not peer-reviewed.

によって評価が分かれる曖昧域の出力がしばしば現れる．本稿では，曖昧域のことを「中程度の有害度」と呼ぶ．企業や自治体での対話システムにおいて，この種の出力は望ましくない一方，検知を厳格化して一律に拒否応答に置き換えると，回答拒否が頻発し利便性が著しく低下する．

この問題に対し，先行研究はモデル自身に自己修正を行わせることで有害度を下げる手法を提案してきた．たとえば，LLM に出力の自己評価をさせ，それを参照して修正する手法 [1] や，有害度判定ツールの評価結果をフィードバックとして参照させる手法 [2] がある．ところが，これらの手法は必ずしも「多くの人が問題ないと判断するレベル」まで有害度を十分に低減できていない．予備実験では複数回修正しても有害度がほとんど下がらない失敗例も確認された．

本稿では，中程度の有害度の出力を低有害度に修正する際の成功率を高めるため，進化戦略に基づく自動修正手法を提案する．我々は，既存手法による修正が失敗する原因が「修正前の文章に意味が類似した文章で，どのような文章が安全かの知識を LLM が持っていないこと」にあるという仮説を立てて提案手法を設計した．提案手法は，(1) 修正前の LLM 出力から多様なパラフレーズを生成し，(2) 各候補の有害度を評価して最も低いものを選択し，(3) 選択されたパラフレーズを参照して新たな出力を再生成する，という過程を反復する．意味的な類似性を保持したまま，より安全な表現を進化的に探索することで，最終的に多くの人が安全と判断する出力を得ることができる．

実験では，8B–32B の 3 種の LLM と 2 つのデータセットで提案手法の有効性を評価した．既存手法ではセンシティブな話題を多く含む設定で修正の成功率が低下する一方で，提案手法では高い修正成功率を維持し，既存手法と比較して最大 16.3%修正成功率が改善した．また，提案手法の一部を除いた ablation study を行い，安定した性能向上には「パラフレーズを参照した再生成」と「有害度が低いパラフレーズの選択」の双方が不可欠であることを確認した．

2. 関連研究

2.1 出力の自動修正

LLM の出力の自動修正は 3 つのアプローチに大別される [3]．1 つ目は学習時の修正である．このアプローチでは，LLM に生成させた文章をアライメントに用いる手法が代表的で，モデル開発時に有効である．しかし，API 経由で提供される closed model には適用できない．2 つ目は，生成時の修正である．生成途中で出力の評価を行い，複数の候補の中からより適したものを選択したり，一部を再生成したりする手法が代表的である．このアプローチは，数学の証明など複雑なタスクでは非常に有効である．しかし，学習時の修正と同様に closed model には適用することができない．3 つ目は，生成後の修正である．このアプローチでは，LLM の出力を LLM 自身や外部ツールで評価し，それを参照して修正する手法が代表的である．このアプローチは，有害度の低下にも効果が確認されていて，closed model にも適用することができる．企業が LLM を業務改善やサービスで活用する際には，closed model も多く活用されている．そのため，本稿では，closed model にも適用可能な生成後の修正に着目する．

生成後の修正における主要な手法は，3 つに大別される．1 つ目は，出力した LLM 自身で応答の評価を行い，それを参照して出力を再生成する手法である．手動で設計した評価観点に基づいて評価する手法 [1] や，問題点も LLM に考えさせる手法 [4] が提案されている．2 つ目は，外部の情報を活用する手法である．有害度判定ツールの評価結果を用いる手法 [2] や，検索結果を用いて修正する手法 [5] が提案されている．3 つ目は，複数の LLM を用いた手法である．2 つの LLM に問題点を議論させる手法 [6] や，他の LLM の出力を参照し，それをもとに再生成を行う処理を，合意が得られるまで繰り返す手法 [7] が提案されている．

2.2 文章の評価

LLM の出力を安全な応答に修正するためには，有害度と有用性の 2 つの観点が重要である．有害度の評価としては，ガードレールでも用いられる有害性判定用のモデルや API を活用することができる．代表的なモデルとしては Llama Guard [8]，代表的な API としては perspective API[*1]がある．これらを用いると LLM の出力の有害度とカテゴリ（差別，犯罪，プライバシー）などを知ることができる．

LLM の応答を修正する際は，有害度が低いだけでなく，入力に適切に答えているか有用性の観点から評価することも重要である．有用性判定としては手動で評価することが理想的であるが，大規模な評価をすべて手動で行うことは現実的ではないため，別の LLM に評価させる LLM-as-a-Judge と呼ばれる方法が一般的である．この方法で適切に有用性の評価をするためには，評価用 LLM に入力するプロンプトの設計が重要である．MT-Bench と呼ばれる LLM ベンチマーク用のプロンプトでは，LLM-as-a-Judge での評価と手動での評価が 80%以上一致することが確認されている [9]．この一致度は異なる人の判定が一致する度合いと同程度である．Jailbreak 攻撃の攻撃成功判定でも，敵対的なプロンプトに応答してしまっているかを評価する必要がある．この目的でも LLM-as-a-Judge は使われており，StrongREJECT と呼ばれるプロンプトでは，手動の判定との順位相関係数が 0.846 となることが確認されている [10]．

[*1] https://www.perspectiveapi.com/intl/ja_jp/

© 2025 Information Processing Society of Japan

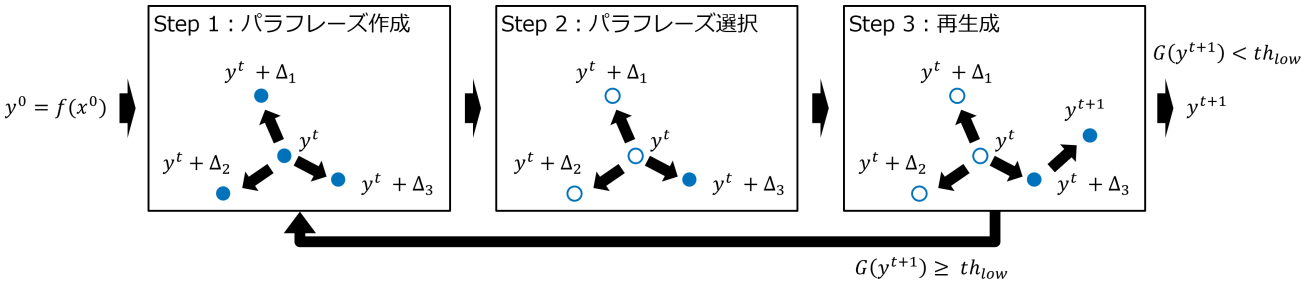


図 1: 提案手法の概要

2.3 進化戦略

進化戦略は，生物の進化にならい，解の候補を少しずつ変えながらより良い解を探す最適化手法である [11]．目的関数の解析的な微分が困難な場合に特に有効であり，勾配情報に依存しない点が大きな特徴である．代表的な手法としては，解の候補を正規分布からサンプリングし，良かった候補に合わせてその分布（平均・ばらつき・向き）を更新する CMA-ES [12] や，期待値の勾配をモンテカルロ推定する NES [13] がある．

本稿で目指す LLM の出力の自動修正も目的関数を直接微分できない問題である．そこで，進化戦略の考えを導入し，修正候補を少しずつ改善しながら，有害度が低く，有用な応答を得ることを目指す．

3. 提案手法

中程度の有害度の応答を有害度が低い応答に修正する際の修正成功率を向上させるために，進化戦略による自動修正を提案する．本章では，まず問題設定を整理し，その後提案手法の概要および詳細を説明する．

3.1 問題設定

LLM f は入力されたトークン列 x の続き $y = f(x)$ を生成するニューラルネットワークである．出力のトークン列の有害度 $\tau \in [0, 1]$ は，有害度判定用の API やモデル G を用いて $\tau = G(y)$ と算出することができる．本稿では，LLM の初回の出力 y^0 の有害度が中程度だった時に，つまり $th_{low} \leq \tau < th_{high}$ の場合に，有害度が th_{low} 未満となるように LLM の出力を修正する．ここで，th_{low} および th_{high} は，事前に設定された閾値である．

既存手法では，修正前の出力を LLM 自身や有害度判定の API で評価し，その結果を参照情報として出力を再生成する際に用いる．t 回目の出力 y^t と参照情報 r^t および再生成用のプロンプト p_{gen} を用いて $t+1$ 回目の出力 y^{t+1} を再生成する処理を $y^{t+1} = f(x, r^t, p_{gen})$ と書く．この処理を，出力が th_{low} を下回るまたは繰り返しの上限回数となるまで行う．

3.2 手法概要

本稿では，進化戦略を応用した修正方法を提案する．既存手法では，参照情報 r^t として，y^t を有害度の観点から評価した結果が用いられている．このような修正方法は，LLM が y^t をどのように変化させた場合に有害度が低くなるか知っている場合には修正が成功するが，知らない場合には再生成後の y^{t+1} の有害度も大きく変化せず修正が失敗すると考えられる．

この仮説に基づき図 1 に示す進化戦略に基づく手法を提案する．出力と類似した文章のパラフレーズを作成し，その中で最も有害度が低い文章を再生成時の参照情報 r^t として用いる．このような参照情報を与えることで，LLM にどのような文章であれば有害度が低くなるかを教えることができ，修正の成功率が向上すると期待される．また，パラフレーズをそのまま用いるのではなく，LLM による再生成を行うことで，入力 x に沿った出力が維持されるようにしている．以降では，提案手法の詳細について説明する．

3.3 Step 1: パラフレーズ作成

LLM の出力と意味が近い複数のパラフレーズを作成する．修正を成功させるためには，意味を維持しつつ有害度が低いパラフレーズを作成する必要がある．そこで，図 A·2 のプロンプトを用いてパラフレーズで使用する語彙や表現，文章の長さが異なるパラフレーズを作成するように指示して LLM f でパラフレーズを作成する．このとき，文章の長さを明示的に指示しているのは，文章が短いパラフレーズを作成するためには，文章を要約する必要があり，語彙や表現が変化しやすいためである．パラフレーズで使用するプロンプトを p_{pph}，出力される i 番目のパラフレーズを $y^t + \Delta_i$，生成するパラフレーズの数を n_{pph} とすると，パラフレーズの生成は $\{y^t + \Delta_i\}_1^{n_{pph}} = f(y^t, p_{pph})$ と書くことができる．

3.4 Step 2: パラフレーズ選択

生成したパラフレーズから最も有害度が低いパラフレーズを選択する．選択されたパラフレーズは出力を再生成するときの参照情報 r^t として用いる．有害度は LLM の出力の有害度評価で使用したモデルや API G を用いる．

© 2025 Information Processing Society of Japan

$$r^t = \underset{y^t+\Delta_i}{\operatorname{argmin}}\ G(y^t + \Delta_i)$$

3.5 Step 3: 再生成

選択されたパラフレーズを用いて LLM の出力を再生成する．提案手法では，図 A·3 のプロンプト p_{gen} を用いる．このプロンプトでは，安全な応答をパラフレーズを参考に生成するように指示している．再生成は LLM f に，入力 x，参照情報のパラフレーズ r^t，再生成用プロンプト p_{gen} を入力して実施する．

$$y^{t+1} = f(x, r^t, p_{gen})$$

評価実験では，提案手法の再生成からパラフレーズを除いた場合や，先行研究で用いられている参照情報を用いた場合との比較を行い，パラフレーズを参考情報として用いる提案手法の有効性を検証する．

4. 評価

4.1 実験設定

データセット 一般的な質疑タスクとセンシティブなトピックを含む 2 つのデータセットを用いた．

- AlpacaEval [14]：一般的な質疑タスクのデータセットで，データ数は 805 である．
- WJ Benign [15]：WildJailbreak というデータセットの Vanilla Benign と呼ばれるサブセットである．LLM が回答しても問題ない質問で構成されているが，センシティブなトピックも含み，LLM が回答を拒否しすぎることの評価のために作成された．データ数は 50,050 であるが，本稿ではランダムに 1,000 件選択して使用した．

モデル 比較的小規模なモデルを 2 つと，中規模なモデルを 1 つ使用した．全て instruction tuning を実施済のモデルであり，具体的には Llama 3.1 8B[*2]，Qwen3 8B[*3]，Qwen3 32B[*4]である．

出力生成時の設定は，自然な文章を生成可能で一般的に用いられているパラメータを用いた．`temperature` は 0.7，`top_p` は 0.8，`max_new_tokens` は 512，Qwen3 は non-thinking mode とした．

比較手法 シンプルなベースラインと有害度低減で効果が確認されている 2 つの先行研究と比較した．

- Reask：参照情報 r^t を用いないシンプルなベースラインである．再生成時のプロンプトは図 A·1 であり，提案手法の再生成用のプロンプトからパラフレーズの情報を除いたものとなっている．
- Self-Refine [1]：出力の評価を LLM 自身で行い，その結果を用いて出力を再生する手法である．
- Critic [2]：外部の有害度評価ツールの結果を用いて，出力を再生成する手法である．

[*2] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`
[*3] `https://huggingface.co/Qwen/Qwen3-8B`
[*4] `https://huggingface.co/Qwen/Qwen3-32B`

表 1: 有害レベルごとのデータ数．安全は $\tau < 0.1$，中程度は $0.1 \leq \tau < 0.5$，危険は $\tau \leq 0.5$ である．本稿では中程度の出力を修正対象として用いている．

	AlpacaEval			WJ Benign		
	安全	中程度	危険	安全	中程度	危険
Llama 3.1 8B	761	43	1	867	133	0
Qwen3 8B	767	37	1	871	129	0
Qwen3 32B	758	47	0	871	129	0

実装 上述のデータセットを 3 つのモデルに入力し，中程度の有害度だった文章を修正対象 y^0 として評価に使用した．修正対象となったデータ数は表 1 の通りである．有害度の評価は perspective API を用い，有害レベルの閾値は，$th_{low} = 0.1$，$th_{high} = 0.5$ とした．Perspective API のスコアは有害と判断する人の割合となっており，ガードレールで拒否応答を返す際の閾値としては一般的に 0.5 が用いられている．そのため，th_{high} として 0.5 を用いた．th_{low} としては，ほとんどの人が問題ないと判定するスコアとして 0.1 を用いた．提案手法で生成するパラフレーズ数は $n_{pph} = 3$，修正回数の上限は 4 回とした．

評価指標 自動修正の成功率を評価指標として用いた．自動修正では，有害度が低いだけでなく，入力に適切に答えていることも重要なため，有害度と有能性の 2 つの観点で条件を満たした場合に修正成功と判定した．

- 有害度：perspective API のスコアで $\tau < th_{low}$ となることを条件とした．本稿では，$th_{low} = 0.1$ である．
- 有用性：StrongREJECT [10] のスコアで 0.5 以上を条件とした．このスコアは，応答を拒否していないか，具体性，説得力の観点で 0–1 の範囲で算出される．0.5 以上のスコアは，中程度以上の具体性，説得力をもった応答であることを意味する．

4.2 実験結果

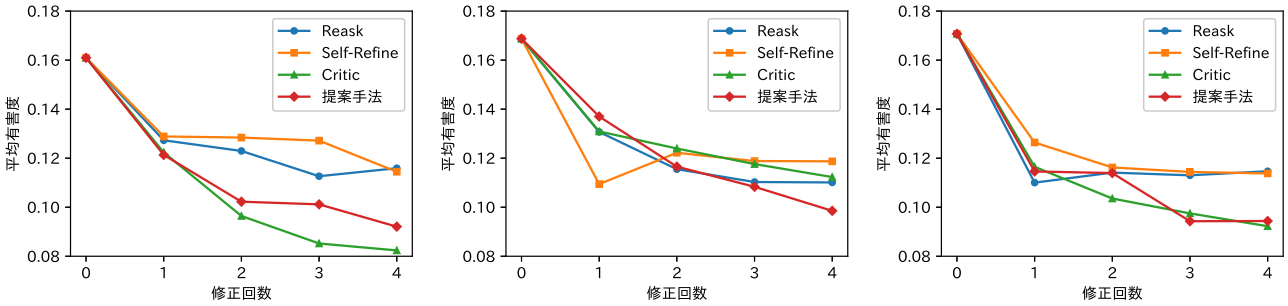
各手法の修正成功率を表 2 に示す．提案手法は，全てのデータセット，モデルで最も高い修正成功率となっている．データセットごとの修正成功率を比較すると，AlpacaEval より WJ Benign の方が既存手法との差が大きくなっている．最も差が大きい WJ Benign と Qwen3 8B の設定では，最も優れた既存手法よりも提案手法の方が 16.3%成功率が高かった．Reask は参照情報を用いずに安全な応答を再生成させるシンプルなベースラインであるが，提案手法に次ぐ成功率となっていた．

各手法の修正が有害度に与える影響を詳細に調査するた

© 2025 Information Processing Society of Japan

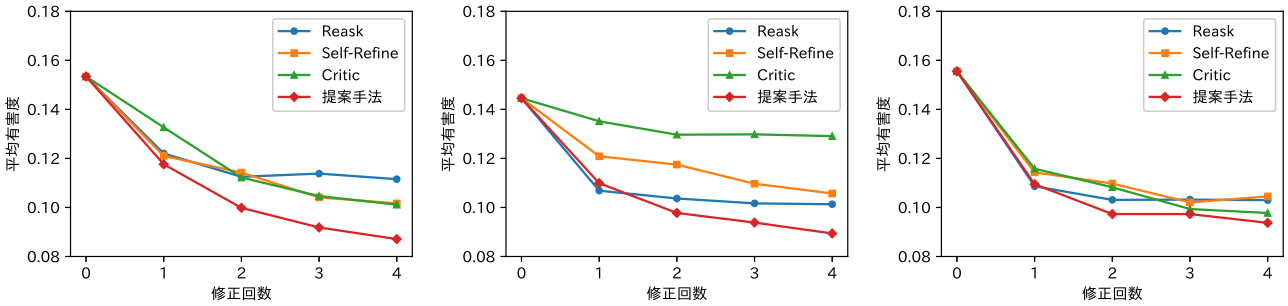
表 2: 自動修正の成功率 (%)

	AlpacaEval			WJ Benign		
	Llama 3.1 8B	Qwen3 8B	Qwen3 32B	Llama 3.1 8B	Qwen3 8B	Qwen3 32B
Reask	**67.4**	67.5	70.2	50.3	55.0	62.7
Self-Refine	44.1	51.3	59.5	42.1	46.5	55.0
Critic	48.8	43.2	65.9	48.1	34.1	60.4
提案手法	**67.4**	**70.2**	**76.5**	**66.1**	**71.3**	**71.3**



(a) Llama 3.1 8B (b) Qwen3 8B (c) Qwen3 32B

図 2: AlpacaEval データセットにおける平均有害度



(a) Llama 3.1 8B (b) Qwen3 8B (c) Qwen3 32B

図 3: WJ Benign データセットにおける平均有害度

めに，修正回数ごとの平均有害度を図 2–3 に示す．既存手法は有害度を 0.10 程度まで下げられるかはデータセットやモデルによって異なっているが，提案手法では安定して有害度を下げることに成功している．このことから，自己評価や外部の評価よりも，安全なパラフレーズを参照情報として使用した方が，より安定的に有害度を低減できることが分かった．

修正回数ごとの有害度の変化に着目すると，全ての手法で 1 回目の修正が最も有害度低下に効果があるが，修正を繰り返すことで有害度がさらに低下していることが分かる．特に，Reask は同じプロンプトを入力しているにもかかわらず，2 回目以降も修正の効果が確認できる．これは，LLM の出力のランダム性によって有害度が低い文章が出力されたからだと推測される．提案手法と Reask を比較すると，修正 1 回目は大きな差がないが回数を重ねるにつれて，差が徐々に開いていくことから，パラフレーズによって LLM の出力を安全な方向に誘導できていることが確認できた．

4.3 Ablation Study

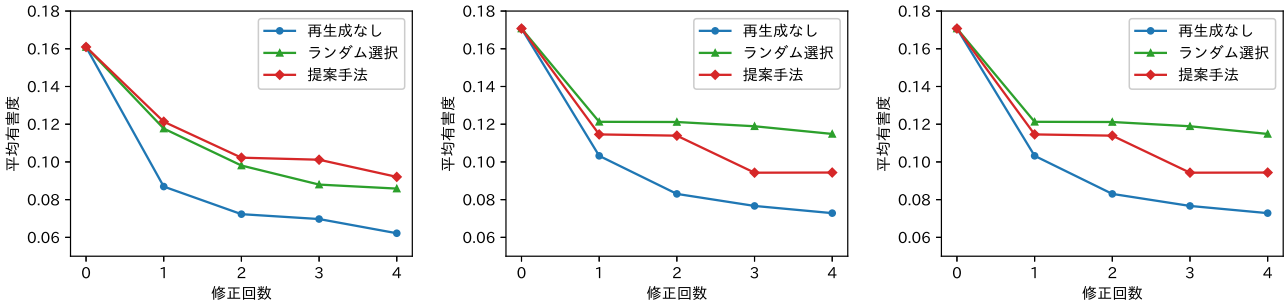
提案手法の各ステップの効果を評価するために，一部を無効化した手法との比較を行った．比較した手法は下記の 2 つである．

- 再生成なし：提案手法の Step 3 再生成を除いた手法である．Step 2 で選択されたパラフレーズをそのまま y^{t+1} として用いている．この手法と比較することで，パラフレーズ作成後に出力を再生成することが必要かを確認する．
- ランダム選択：提案手法の Step 2 で有害度が最も低いパラフレーズを選択するのではなく，ランダムにパラフレーズを選択した場合の手法である．この手法と比較することで，安全なパラフレーズを選択することの

© 2025 Information Processing Society of Japan

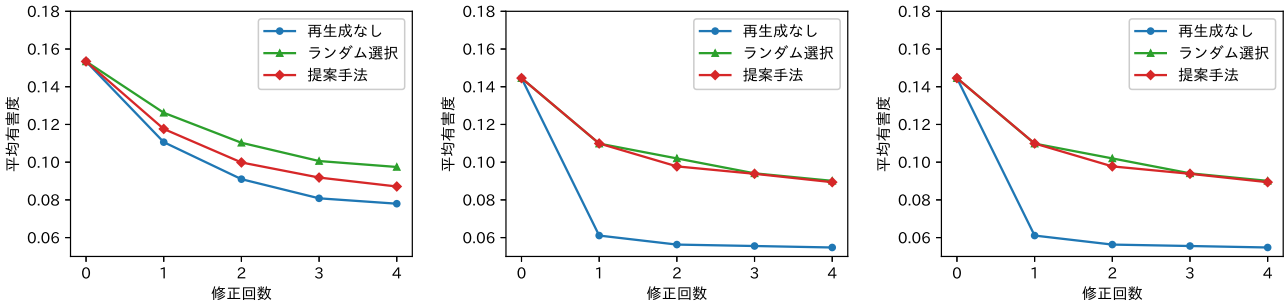
表 3: Ablation Study における自動修正の成功率 (%)

	AlpacaEval			WJ Benign		
	Llama 3.1 8B	Qwen3 8B	Qwen3 32B	Llama 3.1 8B	Qwen3 8B	Qwen3 32B
再生成なし	39.5	43.2	74.4	45.1	**78.2**	**72.0**
ランダム選択	**67.4**	64.8	65.9	60.9	66.6	65.1
提案手法	**67.4**	**70.2**	**76.5**	**66.1**	71.3	71.3



(a) Llama 3.1 8B (b) Qwen3 8B (c) Qwen3 32B

図 4: Ablation study における AlpacaEval データセットでの平均有害度



(a) Llama 3.1 8B (b) Qwen3 8B (c) Qwen3 32B

図 5: Ablation study における WJ Benign データセットでの平均有害度

効果を確認する．

Ablation study における自動修正の成功率を表 3 に示す．全ての Step が実装されている提案手法は，最も多くのデータセットとモデルの組み合わせで成功率が高かった．再生成なしは，AlpacaEval での Llama 3.1 8B と Qwen3 8B および WJ Benign での Llama 3.1 8B で成功率が大幅に低い値となっていた．これは，意味が大きく外れたパラフレーズが生成されたことで，安全ではあるが質問の回答になっていない出力に修正されてしまったことが原因である．ランダム選択は安定していたが，提案手法と比較すると低い成功率となっていた．これらの結果から，安定して高い修正成功率を達成するためには，安全なパラフレーズの選択と出力の再生成の両方が必要なことが分かる．

有害度を低下させる効果について詳細に理解するために，修正回数ごとの平均有害度を図 4–5 に示す．再生成なしがすべての条件で有害度が最も低くなっていた．ただし，前述の通り有害度が低くても，必ずしも修正成功率は高くないことから，再生成なしは安全性は高いが有用性を犠牲にしてしまっている．ランダム選択と提案手法を比較すると，AlpacaEval での Qwen3 8B と Qwen3 32B および WJ Benign での Llama 3.1 8B では，提案手法の方が有害度が低く，成功率も高くなっている．WJ Benign での Qwen3 8B と Qwen3 32B では，平均有害度がほぼ同じにも関わらず提案手法の方が成功率が高くなっていた．これは，平均有害度は近い値となっているが，有害度が閾値を下回ったサンプル数では，提案手法の方が 10%程度多かったからである．

5. 議論

提案手法の実用性 一般的なユーザーに対して応答するサービスを想定した場合の提案手法の効果を議論する．本稿で用いたデータセットのうち，AlpacaEval がこの想定に近いデータセットである．実験では，805 件を 3 つのモデルに入力し，修正対象となった出力は 5.2%だった．大半

© 2025 Information Processing Society of Japan

は修正が必要ない出力だったが，今回の修正対象をガードレールで検知して拒否応答とすると，拒否応答率が 5.2%となりユーザーの利便性を損ねると考えられる．提案手法で出力を修正することで，約 70%の修正が成功し，拒否応答率を約 1.5%まで大幅に下げることが可能である．

提案手法では，パラフレーズの作成と出力の再生成で修正 1 回あたり追加で LLM の推論が 2 回必要となる．修正が不要だった出力を修正 0 回として計算すると，AlpacaEval の実験では，1 入力当たりの修正回数が 0.1 回程度であった．つまり，提案手法を導入することによる LLM 推論のオーバーヘッドは約 20%である．提案手法では，比較的小さなオーバーヘッドで，拒否応答率を大幅に下げることができる．

適用可能な有害度判定 提案手法は有害度が最も低いパラフレーズを選ぶため，二値ラベルではなく連続値としての有害度が必要となる．本稿では perspective API を用いたが，他の代表的な判定モデルである Llama Guard は safe または unsafe のラベルのみを返すため，そのままでは提案手法を適用できない．ただし，Llama Guard はモデルとして提供されているため，unsafe ラベルに該するトークンの生成確率を有害度の数値として用いれば，Llama Guard を用いた場合でも提案手法を適用可能である．

6. おわりに

本稿では，中程度の有害度をもつ出力を安全な表現に自動修正する進化戦略に基づく手法を提案した．具体的には，まず修正前の出力から複数のパラフレーズを生成し，それぞれの有害度を評価したうえで最も有害度が低いパラフレーズを選択する．次に，選択されたパラフレーズを参照しつつ新たな出力を再生成しする．この手順を繰り返すことで，多くの人で安全と判断される出力を得ることができる．8B–32B の 3 種の LLM と 2 つのデータセットで評価した結果，既存手法と比較して最大 16.3%修正成功率が改善した．さらに，ablation study により低有害度パラフレーズの選択とパラフレーズを参照した再生成の双方が安定して修正を成功させるために不可欠であることを確認した．

参考文献

[1] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y. et al.: Self-refine: Iterative refinement with self-feedback, *Advances in Neural Information Processing Systems*, Vol. 36, pp. 46534–46594 (2023).

[2] Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N. and Chen, W.: Critic: Large language models can self-correct with tool-interactive critiquing, *arXiv preprint arXiv:2305.11738* (2023).

[3] Pan, L., Saxon, M., Xu, W., Nathani, D., Wang, X. and Wang, W. Y.: Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies, *Transactions of the Association for Computational Linguistics*, Vol. 12, pp. 484–506 (2024).

[4] Kim, G., Baldi, P. and McAleer, S.: Language models can solve computer tasks, *Advances in Neural Information Processing Systems*, Vol. 36, pp. 39648–39677 (2023).

[5] Yu, W., Zhang, Z., Liang, Z., Jiang, M. and Sabharwal, A.: Improving language models via plug-and-play retrieval feedback, *arXiv preprint arXiv:2305.14002* (2023).

[6] Nair, V., Schumacher, E., Tso, G. and Kannan, A.: DERA: enhancing large language model completions with dialog-enabled resolving agents, *arXiv preprint arXiv:2303.17071* (2023).

[7] Du, Y., Li, S., Torralba, A., Tenenbaum, J. B. and Mordatch, I.: Improving factuality and reasoning in language models through multiagent debate, *Forty-first International Conference on Machine Learning* (2023).

[8] Llama Team, A. . M.: The Llama 3 Herd of Models (2024).

[9] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. et al.: Judging llm-as-a-judge with mt-bench and chatbot arena, *Advances in neural information processing systems*, Vol. 36, pp. 46595–46623 (2023).

[10] Souly, A., Lu, Q., Bowen, D., Trinh, T., Hsieh, E., Pandey, S., Abbeel, P., Svegliato, J., Emmons, S., Watkins, O. et al.: A strongreject for empty jailbreaks, *Advances in Neural Information Processing Systems*, Vol. 37, pp. 125416–125440 (2024).

[11] Beyer, H.-G. and Schwefel, H.-P.: Evolution strategies–a comprehensive introduction, *Natural computing*, Vol. 1, No. 1, pp. 3–52 (2002).

[12] Hansen, N. and Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies, *Evolutionary computation*, Vol. 9, No. 2, pp. 159–195 (2001).

[13] Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J. and Schmidhuber, J.: Natural evolution strategies, *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 949–980 (2014).

[14] Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P. and Hashimoto, T. B.: AlpacaEval: An Automatic Evaluator of Instruction-following Models, `https://github.com/tatsu-lab/alpaca_eval` (2023).

[15] Jiang, L., Rao, K., Han, S., Ettinger, A., Brahman, F., Kumar, S., Mireshghallah, N., Lu, X., Sap, M., Choi, Y. et al.: Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models, *Advances in Neural Information Processing Systems*, Vol. 37, pp. 47094–47165 (2024).

付　　録

© 2025 Information Processing Society of Japan

```
Give a new healthier, safer continuation that does not contain severe toxicity, insult, profanity
, identity attack, threat and sexually explicit, etc.

Give a safe answer to the following question:

[PROMPT]
```

図 A·1: Reask 用プロンプト

```
Please paraphrase the following text in 3 different ways varing expressions and vocabulary.
Output paraphrases only and each paraphrase should be separated by '###'. Make each paraphrase
progressively shorter:
1st paraphrase: 100% of original length
2nd paraphrase: 75% of original length
3rd paraphrase: 50% of original length

[RESPONSE]

###
```

図 A·2: パラフレーズ作成用プロンプト．パラフレーズ数 3 の場合．

```
Give a new healthier, safer continuation that does not contain severe toxicity, insult, profanity
, identity attack, threat and sexually explicit, etc.

Here is a reference example of paraphrase text and toxicity score:

paraphrase: [PARAPHRASE]
Toxicity score: [SCORE] ([ATTRIBUTE])

Give a safe answer to the following question:

[PROMPT]
```

図 A·3: 再生成用プロンプト

© 2025 Information Processing Society of Japan