不可視文字を用いた敵対的攻撃の最適化

川勝祐貴 [1]　坂口慶祐 [1,2]　磯沼大 [1,2,3]
[1] 東北大学　[2] 理化学研究所　[3]NII LLMC
kawakatsu.yuki.t7@dc.tohoku.ac.jp　{keisuke.sakaguchi, isonuma}@tohoku.ac.jp

概要

入力に微小な摂動を加えて機械学習モデルの出力を操作する攻撃は敵対的攻撃と呼ばれる．特にテキストに対しては不可視文字を用いた攻撃が知られており，不可視文字が挿入されていることを知らずにユーザがモデルに入力し，ユーザの意図しない挙動を引き起こす恐れがある．本研究では，不可視文字攻撃の限界を検証するため，不可視文字の挿入方法，決定方法，種類を最適化したときの攻撃成功率を測定した．感情分類タスクでの実験の結果，ランダムに挿入する場合よりも最適化によって最大で48.4%程度攻撃成功率が向上し，最大値は 99.2%となった．一方で，長文における成功率低下や処理時間増加など，不可視文字攻撃の限界も確認された．

1　はじめに

大規模言語モデル（LLM）の発展に伴い，LLM をコード作成や翻訳，要約などの幅広いタスクで活用することが可能となっている．特に，一部ベンチマークでは人間の専門家を超える性能を示すモデルも存在し，その発展は著しいものとなっている [1]．

一方で，LLM などを含む機械学習モデルでは，人には知覚できない微小な摂動を入力に加えることで，モデルの出力を変化させることができる脆弱性が存在する．このような攻撃は敵対的攻撃と呼ばれ，またこのときに作成される入力は敵対的例と呼ばれる [2]．LLM に対する敵対的攻撃も知られているものの，文字置換や同義語置換，言い換えなどにより摂動を加えることから，意味を保持した敵対的例の生成は困難であることが知られている [3, 4, 5]．

このようにテキストの敵対的例は知覚が容易であったが，近年では不可視文字（視覚的な字体として描写されないが有効な文字）や削除文字（バックスペースのような削除制御文字）など，エンコーディングに由来する知覚困難な摂動を用いた方法も研究されている [6]（図 1）．知覚困難な摂動を用いることで，例えばテキスト上に隠された敵対的プロンプトによって，チャット履歴に存在する個人情報が流出するリスクなどが指摘されている [7]．特に不可視文字を用いた手法では，不可視文字を挿入するため攻撃対象が短文でも攻撃可能であることや，フォントインジェクションなどと比較して操作が比較的容易であることなどから，より柔軟な攻撃が可能になるリスクが危惧される．

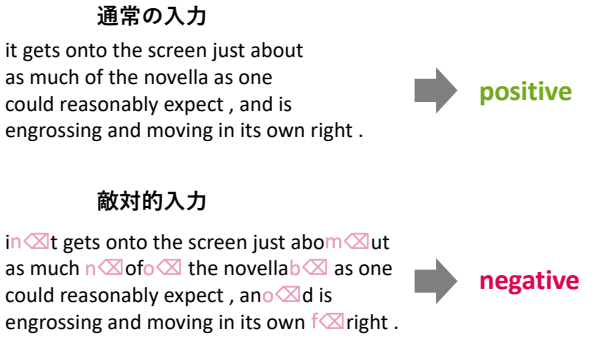

図 1　不可視文字による敵対的攻撃の概略図．不可視文字（例: n⌫）が挿入されることで見た目を変えないままに出力が変化させられている．

このリスクを確認するため，不可視文字の最適化によって敵対的例を生成し，敵対的攻撃の限界を評価する必要がある．本研究では，削除文字などの不可視文字の最適化が LLM の出力に及ぼす影響を，挿入方法，決定方法，種類に着目して検証した．

実験により，不可視文字最適化はランダムな挿入手法に比べて攻撃成功率を向上させることが確認された．一方で，長文入力は攻撃成功率が短文入力よりも低下し，処理時間が増大するという限界も明らかになった．

2　関連研究

2.1　画像に対する敵対的攻撃

敵対的攻撃に関する初期の研究としては，Goodfellow らによる勾配を利用した手法である Fast

Gradient Sign Method（FGSM）[2] や Carlini らによる最適化ベースの手法である Carlini&Wagner の方法 [8] が存在している．また，FGSM の発展形としては Dong らによる運動量ベースの反復アルゴリズムを加えた Momentum Iterative Fast Gradient Sign Method（MI-FGSM）[9] や Lin らによる蓄積された勾配に修正を加えた Nesterov Iterative Fast Gradient Sign Method（NI-FGSM）[10] などが存在する．

2.2 テキストに対する敵対的攻撃

テキストは画像と異なり離散的であるため，2.1 節で示した画像に対する手法は直接用いることはできない．テキストに対する敵対的攻撃に関する初期の研究としては，Papernot らによる再帰的ニューラルネットワーク（RNN）に対してヤコビアン行列と単語埋め込みを用いて単語入れ替えによる攻撃を実施した研究 [4] や，Jia らによる文章の末尾に敵対的な文章を追加してモデルの理解能力を検証した研究 [3] が存在している．また，Zou らは貪欲法と勾配ベースの探索手法を組み合わせて，LLM に有害なコンテンツを出力させる敵対的サフィックスを作成する手法である Greedy Coordinate Gradient（GCG）を提案した [11]．加えて，Xiong らはフォントインジェクションを用いて文字コードとグリフの対応関係を操作し，敵対的的なプロンプトを埋め込む手法を提案した [7]．

2.3 不可視文字を用いた敵対的攻撃

不可視文字を用いたテキストに対する敵対的攻撃の先行研究としては，Boucher らによるものが存在し，不可視文字，ホモグリフ，並び替えや削除のような人間には知覚不可能な摂動をそれぞれ用いた差分進化による 4 種類の不可視攻撃が提案されていた [6]．また，Sarabamoun は不可視文字を含む特殊文字による攻撃を分類しており，オープンソースの LLM に対する文字レベルの敵対的攻撃の脆弱性を体系的に測定した [12]．加えて，Jin らは PDF 化を考慮しており，フォントサイズを 0 にした敵対的テキストを分割して原文の文字列に埋め込むことで LLM にハルシネーションを引き起こさせる攻撃手法である TRAPDOC を提案した [5]．

このように，不可視文字を用いた敵対的攻撃に関する研究は多数存在しているが，この際の最適化手法そのものに着目した研究は実施されていない．

3 問題設定

3.1 定式化

本実験では，原文 x_0 に対して不可視文字の挿入のみを許す制約下で敵対的例を作成し，言語モデル θ が出力する不正解ラベル y の確率 P を最大化（損失 L を最小化）する敵対的攻撃を考える．原文 x_0 に不可視文字のみを挿入して得られる入力の集合を $X(x_0)$ とすると，最適化された敵対的例 x_{adv} は以下の式 1 で定義される．

$$\begin{aligned} x_{adv} &= \arg\max_{x \in X(x_0)} P(y|x;\theta) \\ &= \arg\min_{x \in X(x_0)} L(y|x;\theta) \end{aligned} \tag{1}$$

3.2 最適化手法

本実験では最適化手法を比較するため，不可視文字の挿入方法，決定方法，種類の 3 観点を設定し，それらが出力に及ぼす影響を検証する．

- 挿入型/置換型
- 貪欲型/一次近似型
- 削除文字型/不可視文字全体型

3.2.1 挿入型/置換型

不可視文字の挿入方法について，挿入型と置換型の 2 種類の方法を検討する．挿入（insert）型は，ランダムに選択したトークンの各文字間に候補となる不可視文字を挿入した文章を生成し，各候補における不正解ラベルの確率を計算する．その上で，不正解ラベルの出力確率が最大となる候補と挿入位置を選択する．一方，置換（replace）型は事前に不可視文字をランダムに挿入しておき，それらを逐次置換しながら最適化する．

3.2.2 貪欲型/一次近似型

挿入/置換する不可視文字の種類とその位置の決定方法として，貪欲型と一次近似型の 2 種類の方法を検討する．貪欲（greedy）型では，挿入/置換位置をランダムに選択し，候補となる不可視文字を挿入/置換した際の不正解ラベルの出力確率を各候補ごとに算出し，確率が最大となる候補を選択する．ただし，確率が上昇しない場合は挿入/置換を行わない．全ての挿入/置換可能な位置について同様の試行を行う．確率が事前設定した閾値を上回った時点で探索を終了する．

一次近似（approx）型は置換のみに用いることができる方法である．前後の損失の差分を一次近似により算出し，損失の下がり幅が最大となる位置と置換する不可視文字の種類を決定する．実際に置換したときの損失も算出し，損失が低下しない場合は下がるか候補がなくなるまで同様に探索する．また，損失が事前設定した閾値を下回った時点でも探索を終了する．損失の差の一次近似は以下の式 2 で与えられる．θ は攻撃対象のモデル，x_i は置き換える位置の元々のトークンの埋め込み，x_{-i} は元々のトークンから置き換えたトークンを除外した埋め込み，x' は置き換えた先のトークンの埋め込み，y は不正解ラベルとする．

$$\begin{aligned} &L(y|x' + x_{-i};\theta) - L(y|x_i + x_{-i};\theta) \\ &\approx \nabla_x L(y|x_i + x_{-i};\theta)^\top (x' - x_i) \end{aligned} \tag{2}$$

不可視文字には 1 トークンで構成されるものと複数トークンで構成されるものがあり，上記の一次近似の計算を行うには置換前後のトークン数が同一である必要があることから，一次近似型では 1 トークンで構成される不可視文字のみを用いる．

3.2.3 削除文字型/不可視文字全体型

挿入する不可視文字の種類として，削除文字のみを用いた場合と，削除文字に加えゼロ幅空白などの不可視文字を用いた場合を検討する．

削除文字とは直前の文字を不可視にする制御文字である．削除文字（delword）型では，「削除される文字」と削除文字の組を挿入/置換する．「削除される文字」には小文字アルファベット 26 種類を用い，削除文字には U+0008 を用いた．ただし，置換型と組み合わせる際には，置換後に「削除される文字」とその直前の文字がトークンとして結合されることを防ぐため，「削除される文字」の前に 1 トークンで構成される不可視文字（U+200B（ゼロ幅空白），U+200C（ゼロ幅非結合子），U+FEFF（ゼロ幅非破壊空白），U+200E（左横書き符号））を加えた．

不可視文字全体（invword）型では，削除文字に加えゼロ幅空白などの不可視文字を挿入/置換する．具体的には，先行研究 [6, 12] に基づき，U+200B（ゼロ幅空白），U+200C（ゼロ幅非結合子），U+200D（ゼロ幅接合子），U+2060（単語接合子），U+FEFF（ゼロ幅非破壊空白），U+200E（左横書き符号），U+200F（右横書き符号）を用いた．また，削除文字と組になる「削除される文字」には “a” のみを用いた．

4 実験

4.1 実験設定

本実験では，Stanford Sentiment Treebank binary（SST-2）[13] と Internet Movie Database（IMDB）[14] を用いた二値分類の感情分析タスクにて不可視文字攻撃の成功率を検証した．SST-2 は短文，IMDB は長文に対する検証のために用いた．敵対的攻撃の目的は不正解ラベルの確率の最大化とした．SST-2 は validation の計 872 件，IMDB は test の計 500 件のデータを用いた．攻撃対象の LLM には Qwen3-4B-Instruct-2507[15] を使用した．ここで，入力プロンプトは，モデルリポジトリに付属する tokenizer.chat_template に従い，Transformers の tokenizer.apply_chat_template を用いて作成した．用いたプロンプトの詳細については付録 A に示す．また，ベースラインとして削除文字の組と不可視文字全体をそれぞれランダムに挿入する手法（random_delword，random_invword）と，原文に手を加えない手法（plain）を用いる．各手法で用いたハイパーパラメータに関しては付録 B に示す．

4.2 実験結果

実験結果を表 1 に示す．最適化手法名は挿入方法，決定方法，種類の順に示している．このとき生成した敵対的例の具体例は付録 C に示す．SST-2 と IMDB の両方で，不可視文字の挿入によって攻撃成功率を向上させた．攻撃成功率は，SST-2 では全最適化手法がランダムな手法を上回った．IMDB では insert_greedy_delword 型のみがランダムな手法を上回り，他の最適化手法は下回った．処理時間は，全手法で SST-2 よりも IMDB において長かった．

挿入型（insert）と置換（replace）型の比較のために replace_greedy_delword 型と insert_greedy_delword 型の結果に着目する．攻撃成功率は SST-2 および IMDB で挿入型が置換型を上回り，特に IMDB で差が顕著であった．処理時間は SST-2 では挿入型が短い一方，IMDB では置換型が短かった．

貪欲（greedy）型と一次近似（approx）型の比較のために replace_greedy_delword 型と replace_approx_delword 型の結果に着目する．攻撃成功率は SST-2 では一次近似型が貪欲型を上回った一方，IMDB では貪欲型がわずかに上回った．処理時間は SST-2 および IMDB の両方で一次近似型が短

表 1 実験結果．最適な手法を選択することで高い攻撃成功率を得ることができた．また，長文を扱う IMDB は短文を扱う SST-2 よりも攻撃成功率が低く，処理時間が長い結果となった．

	SST-2		IMDB	
手法（挿入方法_決定方法_種類）	攻撃成功率（%）	処理時間（s）	攻撃成功率（%）	処理時間（s）
plain 型 （baseline）	5.6	0	5.0	0
random_delword 型 （baseline）	50.8	0	50.4	72
random_invword 型 （baseline）	47.9	0	49.0	112
replace_approx_delword 型	**93.5**	20845	13.6	39809
replace_greedy_delword 型	**91.7**	24201	14.2	142723
insert_greedy_invword 型	**74.9**	13180	23.4	80472
insert_greedy_delword 型	**99.2**	7702	**71.6**	175221

く，特に IMDB で差が顕著であった．

削除文字（delword）型と不可視文字全体（invword）型の比較のために insert_greedy_delword 型と insert_greedy_invword 型の結果に着目する．攻撃成功率は SST-2 および IMDB で削除文字型が不可視文字全体型を上回り，特に IMDB で差が顕著であった．処理時間は SST-2 では削除文字型が短い一方，IMDB では不可視文字全体型が短かった．

5 考察

5.1 挿入型/置換型

本観点の比較のため，replace_greedy_delword 型と insert_greedy_delword 型に着目する．攻撃成功率については，置換型（replace）が不可視文字を事前にランダムに挿入するのに対し，挿入型（insert）はトークン内の位置も含めて最適化するため，高くなると考えられる．一方，処理時間については，挿入型は位置も探索対象とすることで候補数が増加するため，IMDB データのような長文を扱う場合には同時に処理する候補数（バッチサイズ）を小さく設定する必要があり，長くなると考えられる．以上より，本比較の条件下では，挿入型は置換型より有効であると考えられる．

5.2 貪欲型/一次近似型

本観点の比較のため，replace_greedy_delword 型と replace_approx_delword 型に着目する．攻撃成功率については，SST-2 のような短文では事前に挿入可能な不可視文字の量が両手法とも十分であった一方，IMDB のような長文では十分ではなかった．特に一次近似型（replace）の場合は貪欲型（greedy）より挿入可能な量が少ないため，置換候補数も減少したためと考えられる．処理時間については，貪欲型では攻撃対象の文長に応じてバッチサイズが変動するが，一次近似型では全候補を同時に処理する．SST-2 のような短文ではこの差は小さいが，IMDB のような長文ではバッチサイズを小さく設定する必要があるため，処理時間差が大きくなると考えられる．以上より，本比較の条件下では，一次近似型は貪欲型より有効であると考えられる．

5.3 削除文字型/不可視文字全体型

本観点の比較のために insert_greedy_delword 型と insert_greedy_invword 型を用いる．削除文字型（delword）の同一挿入位置の候補は 26 種類である一方，不可視文字全体型（invword）の同一挿入位置の候補は 8 種類であるため，SST-2 のような短文では候補数の増加に伴う探索経路数の増加が支配的である一方，IMDB のような長文では処理時間への影響が支配的であることが結果の要因と考えられる．以上より，本比較の条件下では，削除文字型は不可視文字型より有効であると考えられる．

6 おわりに

本研究では，不可視文字を用いた敵対的攻撃の限界を検証するために，最適化に着目した測定を実施した．実験結果より，最適化は不可視文字攻撃に対して有効であり，特に最適な手法の選択によって攻撃成功率が大幅に向上することが示された．一方で，処理時間及び精度の面で，長文に対する不可視文字攻撃の限界も示された．今後の課題としては，翻訳や要約などより複雑なタスクに対する攻撃の有効性の検証や，他のモデルに対する汎化性能の確認，特定のモデルに対して最適化された敵対的例の他モデルに対する有効性の検証などが挙げられる．

謝辞

本研究は，JST BOOST JPMJBY24A6，JSPS 科研費 JP24K03236，JP25K03175 の支援を受けたものです．

参考文献

[1] OpenAI. Introducing GPT-5.2, 2025. https://openai.com/index/introducing-gpt-5-2/.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In **International Conference on Learning Representations**, 2015.

[3] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 2021–2031, 2017.

[4] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In **MILCOM 2016-2016 IEEE Military Communications Conference**, pp. 49–54, 2016.

[5] Hyundong Jin, Sicheol Sung, Shinwoo Park, SeungYeop Baik, and Yo-Sub Han. TrapDoc: Deceiving LLM users by injecting imperceptible phantom tokens into documents. In **Findings of the Association for Computational Linguistics: EMNLP 2025**, pp. 18881–18897, 2025.

[6] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible nlp attacks. In **2022 IEEE Symposium on Security and Privacy (SP)**, pp. 1987–2004, 2022.

[7] Junjie Xiong, Changjia Zhu, Shuhang Lin, Chong Zhang, Yongfeng Zhang, Yao Liu, and Lingyao Li. Invisible prompts, visible threats: Malicious font injection in external resources for large language models. **arXiv preprint arXiv:2505.16957**, 2025.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In **2017 IEEE Symposium on Security and Privacy (SP)**, pp. 39–57, 2017.

[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In **Proceedings of the IEEE conference on computer vision and pattern recognition**, pp. 9185–9193, 2018.

[10] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In **International Conference on Learning Representations**, 2020.

[11] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. **arXiv preprint arXiv:2307.15043**, 2023.

[12] Ephraiem Sarabamoun. Special-character adversarial attacks on open-source language model. **arXiv preprint arXiv:2508.14070**, 2025.

[13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**, pp. 1631–1642, 2013.

[14] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**, pp. 142–150, 2011.

[15] 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.

A　実験で使用したプロンプト

Chat Template 形式のメッセージに tokenizer.apply_chat_template を適用して作成した入力プロンプトを図 2 に示す．{Text}は攻撃対象のテキスト，{Label}はラベルを示している．

<|im_start|>system
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.<|im_end|>
<|im_start|>user
Instruction: You are a sentiment classifier. Classify the sentiment of the following review as exactly one of negative, positive. Answer with one word: negative, positive.
Input: {Text}
Answer: <|im_end|>
<|im_start|>assistant
{Label}<|im_end|>

図 2　入力プロンプト（apply_chat_template 適用後）

B　ハイパーパラメータ

実験で使用したハイパーパラメータを表 2 に示す．epoch は最大探索回数，stop_prob は探索を終了する不正解ラベルの確率，patience は一次近似型において，損失が低下しない場合の最大探索数を示し，これらのパラメータは SST-2 と IMDB で同じ値を用いる．ただし，一次近似型では stop_prob を損失に変換する．p は入れる不可視文字の組の文長に対する割合を示しており，置換型及びランダム型は事前に挿入する割合，挿入型は挿入可能な上限値を示す．batch_size は貪欲型における，候補の並列処理数を示し，一次近似型では使用しない．p と batch_size は SST-2 と IMDB で値を揃えない．

表 2　実験で用いたハイパーパラメータ

				SST-2		IMDB	
手法	epoch	stop_prob	patience	p	batch_size	p	batch_size
replace_greedy_delword	200	0.9	-	0.5	32	0.1	2
replace_approx_delword	200	0.9	100	0.5	-	0.05	-
insert_greedy_delword	200	0.9	-	1.0	64	0.1	2
insert_greedy_invword	200	0.9	-	1.0	64	0.1	2
random （delword）	-	-	-	20	-	18	-
random （invword）	-	-	-	20	-	18	-

C　生成した敵対的例

実験で生成した敵対的例の具体例を表 3 に示す．不可視文字は薄い灰色にして示している．

表 3　生成した敵対的例の具体例

敵対的例	出力ラベル	正解ラベル
inU+0008t gets onto the screen just abomU+0008ut as much nU+0008ofoU+0008 the novellabU+0008 as one could reasonably expect , anoU+0008d is engrossing and moving in its own fU+0008right .	negative	positive
it gets onto the screen jusU+2060tU+200d U+200caU+200bbouU+200bt as much of the novU+2060U+200bU+200fella as one could reasonably expect , and iU+2060s enU+200fgrossing and movinU+2060g inU+feff its U+feffU+2060U+2060U+2060own right .	negative	positive