2025 年度 iTL 先端的プロジェクト奨学金中間報告書

日本語リポグラム自動生成のための自然言語処理技術の開発

中央大学 国際情報学部

22G1104002B 橋本葵

指導教員：飯尾淳

提出日：2025 年 11 月 29 日

目次

第 1 章　研究の概要・背景 4
1.1　研究背景 4
1.2　研究目的 4

第 2 章　関連研究 6
2.1　制約付き文章生成とスタイル変換 6
2.2　リポグラム研究と日本語の空白 6
2.3　LLM と制約付き生成 7

第 3 章　研究の課題と評価指標 8
3.1　タスク定義と研究の課題 8
3.1.1　制約遵守（Constraint） 8
3.1.2　意味保持（Semantic Preservation） 8
3.1.3　文法性・自然性（Fluency） 9
3.1.4　研究の課題 10
3.2　評価指標と測定項目 10
3.2.1　主指標 10
3.2.2　補助指標（書き換えの性質・品質） 10
3.3　比較対象となる生成方式 11

第 4 章　提案する手法とシステム構成 12
4.1　リポグラム生成フレームワークの概要 12
4.2　oneshot/sequential 方式 14
4.3　Web アプリケーション実装 15

第 5 章　実験の設定と結果の概要 16
5.1　データセット 16
5.2　実験条件と評価指標 16
5.3　制約遵守率と方式比較 17
5.4　VRR・TTR とエラー分析 19

第 6 章　今後の計画 21
6.1　実験拡張の方針 21
6.1.1　制約タイプの拡張 21
6.1.2　文ジャンル・文長の多様化 21
6.1.3　反復実験と統計的検定の高度化 21
6.2　主観評価と Web アプリケーションの活用 21

6.2.1 評価プロトコルの設計 21
6.2.2 Web アプリ側の機能拡張 22
6.3 精度改善と応用領域の検証 22
6.3.1 精度改善のためのアプローチ探索 22
6.3.2 応用領域の検証 23
第 7 章 まとめ・期待される成果 25
7.1 本研究の現時点でのまとめ 25
7.2 期待される成果と意義 25
7.3 今後の課題 25

第1章　研究の概要・背景

本報告書は、2025 年度 iTL 先端的プロジェクト奨学金の採択課題「日本語におけるリポグラム自動生成のための自然言語処理技術の開発」について、中間報告書として整理したものである。まず、本章で研究の目的と背景を述べる。

本プロジェクトの成果は卒業論文にも反映される予定であるが、本文書はあくまで奨学金プロジェクトの進捗報告を主眼とし、研究計画書で示した方針を踏まえつつ、現在までに得られている知見と進捗をまとめたものである。

1.1　研究背景

本研究は、日本語において特定の文字を一切含まない文章「リポグラム」を自動的に生成する手法を検討することを目的とする。リポグラムは、もともと特定の文字を意図的に用いないという制約のもとで書かれた文学作品や言語遊びとして知られており、アルファベット言語では「e」を使わない小説などの著名な事例が存在する。しかし、日本語において同様の制約を課すと、助詞や活用語尾、頻出語彙に禁止文字が含まれやすく、単純な文字除去や置換では文法的・意味的な破綻が生じやすい。そのため、日本語リポグラムは人手で作成する場合にも高度な言語能力を要し、自動生成はさらに困難な課題となる。

一方で、近年は大規模言語モデル（Large Language Model; LLM）の発展により、機械による自然な日本語文生成が実用レベルに達しつつある。敬語変換やポジティブ表現への書き換え、NG ワード除去など、さまざまな制約付き文章生成タスクに LLM を応用する試みも進んでいる。しかし、禁止単語やスタイル変換と比較して、特定文字を完全に排除した上で意味と自然さを保つリポグラム生成は、制約の粒度がより細かく、文全体の構造を再設計する必要がある場合も多い。そのため、プロンプトで「この文字を含めないでください」と指示するだけでは十分な制御が効かないことが、簡単な試行や先行事例から示唆されている。

1.2　研究目的

前述した背景から、本研究では、日本語リポグラム生成を「制約遵守」「意味保持」「文法性」の三つの観点で評価しながら、LLM を用いた書き換えフレームワークを設計・比較することを目指す。具体的には、文全体を一度に言い換える「一発生成（one-shot）」方式と、文脈を利用しつつトークンや文節単位で逐次的に書き換えていく「逐次生成（sequential）」方式の二つを用意し、それぞれがどの程度禁止文字を回避しつつ、元文の意味や自然さを保てるかを定量的に評価する。特に、日本語特有の要因として、助詞や活用語尾に禁止文字が含まれる場合や、「あ行」など行単位で複数文字をまとめて禁止する場合など、文構造への影響が大きい条件に注目する。

本研究の最終的な目的は、日本語リポグラムという一見特殊な課題を通じて、「細かい制約を課しながら自然な文章を生成する」という一般的な問題設定に対して知見を得ることである。リポグラム生成の枠を超えて、NG ワードを安全に避けながら意味を保つチャット応答や、発音しづらい音を含まない代替表現の生成、言語教育における語彙・文法意識を高める練習問題の自動生成など、応用可能性は広い。本中間報告では、これまでに構築した実験環境と比較実験の初期結果を整理し、Web アプリケーション化と定量評価の二つの軸から進捗を報告するとともに、今後の本格的な実験および卒業論文執筆に向けた課題と計画を示す。

第2章　関連研究

本章では、本研究が位置づけられる制約付き文章生成やスタイル変換の研究動向を整理した上で、英語・フランス語におけるリポグラム生成の先行事例と、日本語リポグラムに特有の課題について概観する。さらに、LLM に対する負の語彙制約や制約付きデコーディングに関する研究を取り上げ、本研究のアプローチとの関係と違いを明確にする。

2.1　制約付き文章生成とスタイル変換

本研究は、広い意味での「制約付き文章生成（constrained text generation）」の一種として位置づけられる。制約付き生成に関する先行研究としては、NG ワードや不適切表現を含むトークン列を出力から排除する手法や、敬語変換やポジティブ表現へのスタイル変換、特定属性（感情・文体・話者など）を付与する条件付き生成モデルなどが挙げられる。これらの研究では、既存の文を入力として受け取り、意味をある程度保持しつつ、所望の制約やスタイルを満たす出力に変換するという点で、本研究と共通する側面を持つ。

一方で、NG ワード除去などのタスクでは、禁止語のリストが単語レベルで管理されることが多く、単純な置換やマスクといった処理でも一定の効果が得られるのに対し、本研究で扱うリポグラムは文字レベル・音レベルの制約であり、文全体への影響がより広範である点が異なる。

スタイル変換の文脈では、ニューラルネットワークを用いて入力文を潜在表現に写像し、そこから異なるスタイルの文を復元する手法や、LLM に対しプロンプトを工夫することで敬語・カジュアル体・ポジティブ表現などへの変換を行う手法が提案されている。例えば，Shen ら（2017）は非並列データからのスタイル変換を目的として潜在表現のアラインメントに基づく手法を提案し，Li ら（2018）は元文からスタイル情報を削除し，参照文からスタイルを取得し，再生成するという「Delete–Retrieve–Generate」型の枠組みを示している。

これらの研究は、意味保持とスタイル変化のトレードオフをどのように制御するか、評価指標としてどのような自動スコアや人手評価を用いるか、といった議論を蓄積しており、本研究における「意味保持」と「制約遵守」のバランス設計に多くの示唆を与える。ただし、スタイル変換では「文のどこか一部が変わればよい」ケースも多いのに対し、リポグラムでは出力全体から禁止文字を完全に排除する必要があり、制約違反は一点でもあれば失敗とみなされる。その意味で、本研究はより厳格な制約条件を扱うタスクと言える。

2.2　リポグラム研究と日本語の空白

リポグラムそのものに関する先行研究は、英語やフランス語などアルファベット言語を対象とした例が中心である。たとえば、「e」抜き小説のような極端なリポグラム作品を分

析した文学研究や、特定の文字を含まない単語列を自動生成するプログラムが、趣味的プロジェクトやパズルとして公開されている。また、GitHub 上には単語レベルの置換や辞書ベースの探索によって英語リポグラム文を生成する実装例も存在し、フランス語における「e」抜き小説を対象に GPT 系モデルをファインチューニングしてリポグラム文を生成するプロジェクト（lipogram_e）も報告されている。

しかし、これらはアルファベットのように文字と音の対応が比較的単純な言語を前提としており、日本語への直接的な適用は難しい。日本語では、漢字・ひらがな・カタカナが混在し、同じ音に複数の表記が存在するだけでなく、形態素の境界も明示的には現れないため、単純な文字列操作では意味を保った生成が困難である。

日本語におけるリポグラム研究や実装例は、現時点ではほとんど体系的に整理されていない。個人のブログや SNS 上で、特定の仮名を使わずに文章を書く試みが散発的に見られるものの、データセットや評価指標を整備した上で自動生成手法を比較検討した研究はほとんど見当たらない。本研究は、この空白を埋める試みとして、日本語リポグラム生成のためのタスク定義・評価指標・実験プロトコルを提示し、LLM を用いた複数の生成方式を比較する点に新規性がある。また、制約チェックにおいて読みベースの判定を導入することで、表記の違いを越えて禁止音を検出する仕組みを実験的に検証する点も、日本語ならではの貢献と位置づけられる。

2.3 LLM と制約付き生成

LLM に対する制約付与の観点からは、negative constraints や制約付きデコーディングに関する研究との関連も指摘できる。近年、一部の研究ではデコーディング時に特定トークン列を禁止する手法や、制約充足を保証する探索アルゴリズムが提案されている。たとえば Kajiwara (2019) は、パラフレーズ生成において言い換え対象語を事前に同定した上で、その語を出力してはならない負の語彙制約（negative lexical constraints）をデコーディングに組み込む手法を提案し、必要な箇所のみを確実に書き換えつつ意味保持と流暢さを両立できることを示している。

しかし、商用 API として提供される LLM に対してこれらの制約付きデコーディングをそのまま適用することは難しい場合が多い。本研究では、API ベースの LLM を前提としつつ、プロンプト設計と生成後の再書き換えパイプラインによって制約を満たす実用的な手法を模索する。その意味で、本研究はアルゴリズムレベルの理論的貢献というよりも、実験的な比較と設計指針の提示に重きを置いた応用研究として位置づけられる。

第3章　研究の課題と評価指標

本章では、本プロジェクトで扱うリポグラム生成タスクを形式的に定義し、そこから導かれる研究の課題と評価指標を整理する。これにより、第 4 章以降で述べる提案手法や実験設定が、どのような観点から評価されるべきかを明確にする。

3.1　タスク定義と研究の課題

本研究で扱うタスクは、「与えられた日本語文と禁止文字集合に対して、その禁止文字を一切含まない日本語文を生成すること」である。入力としては、元となる日本語文（ことわざ、小説の一文、日常文など）と、禁止する文字の集合を与える。禁止文字は、ひらがな一文字の集合として指定する場合（例：「い,さ」）に加え、「あ行」のように行単位で複数の仮名をまとめて禁止する場合も想定する。アルゴリズムは、この入力に対して禁止文字を含まない新しい文を出力するが、出力文が単に禁止文字を避けていれば良いのではなく、日本語として自然であり、元文の意味内容を大きく損なっていない必要がある。

形式的には、入力文を(x)、禁止文字集合を(B)、出力文を(y)としたとき、次の条件を満たす(y)を生成することが目標となる。

3.1.1　制約遵守（Constraint）

制約遵守とは、出力文(y)に含まれるすべての文字（あるいは読み）が、禁止集合(B)に属さないことをいう。文字ベース制約では表記上の文字列、読みベース制約では形態素解析により得られた読みの仮名列に対してこの条件を課す。

たとえば、元文xが「彼は朝ごはんを食べなかった。」で、禁止文字集合Bが「あ」のとき、次のような出力は制約違反となる。

y^1:*「彼は朝飯を食べなかった。」*

これに対して、意味を保ちつつ「あ」を避けた出力の一例は以下のような文である。

y^2:*「彼は朝食を食べなかった。」*

ここでは「朝ごはん(あさごはん)」を「朝食(ちょうしょく)」に言い換えることで、「あ」を含まない表現に置き換えている。

3.1.2　意味保持（Semantic Preservation）

意味保持とは、元文(x)と出力文(y)の意味的類似度が十分に高いことをいう。完全な同義である必要はないが、元文が持つ主要な情報や含意が失われていないことが望ましい。

たとえば、元文 x が「彼女は東京でエンジニアとして働いている。」に対して、

y^1:*「彼女は東京で技術者として勤務している。」*

は「エンジニア／技術者」「働いている／勤務している」といった語彙の違いはあるもの

の、場所（東京）と職種（エンジニア）という主要な情報が保たれており、意味保持されているとみなせる。

一方で、以下の出力文は意味保持しているとはみなせない。

y^2:*「彼女は大阪で学生生活を送っている。」*

この出力文は、勤務地が東京から大阪に変わり、職業もエンジニアから学生に変わっているため元文の意味内容から大きく逸脱しており、意味保持がなされていない例といえる。

3.1.3　文法性・自然性（Fluency）

文法性・自然性とは、出力文(y)が日本語として自然に読めることをいう。多少のスタイル変化（です・ます体から常体への変化など）は許容するが、意味が理解しづらいほどの文法破綻は避ける。

たとえば、元文 x「この本はとてもおもしろかったです。」に対して、

y^1:*「この本はとてもおもしろかった。」*

は、丁寧体から常体へのスタイル変化はあるものの、日本語として自然であり、文法性・自然性の観点からは問題ないとみなせる。

一方で、以下の出力文は日本語として自然とはいえない。

y^2:*「本とてもおもしろいでした。」*

この出力文は、語順や活用が不自然で、母語話者にとって違和感の大きい文は、たとえ禁止文字を避けていても、文法性・自然性を満たしていない例である。

3.1.4　研究の課題

このタスク定義に基づき、本プロジェクトで特に検証したい研究課題（Research Question; RQ）は次の三点である。

1. RQ1: 日本語リポグラム生成において、逐次生成方式（sequential）は一発生成方式（oneshot）に比べて、どの程度制約遵守率を改善できるか。
2. RQ2: 制約遵守率を維持・向上させたうえで、元文の意味保持や自然さをどの程度保てるか。その際、両方式の語彙置換率（Vocabulary Replacement Rate; VRR）や語彙多様性（Type-Token Ratio; TTR）にはどのような差が生じるか。
3. RQ3: 制約タイプ（単音禁止/行禁止など）や文長・文ジャンルの違いによって、両方式の得意・不得意はどのように変化するか。

3.2　評価指標と測定項目

上記の研究課題を検証するため、本研究では「制約をどの程度厳格に守れるか」「どの程度まで元文の意味と自然さを保てるか」「現実的な時間内で動作するか」という観点から、二つの方式を多面的に比較する。そのために、いくつかの定量指標を組み合わせて生成結果を評価することとし、データセットや具体的な実験条件については第 5 章で詳述し、ここでは各指標の役割に焦点を当てる。

3.2.1　主指標

まず主指標として、制約遵守率（Constraint Satisfaction Rate）を用いる。これは、生成文の読みが禁止かなを一切含まない文の割合を表す指標であり、リポグラムとして成立しているかどうかを判定する最も重要な基準である。逐次生成方式と一発生成方式のどちらがより高い制約遵守率を達成できるかを比較することで、RQ1 に対応する検証を行う。

例えば、ある条件下で生成した文の総数を(N)、そのうち読みベースのチェックで制約違反が一度も検出されなかった文の数を(N_{success})とすると、制約遵守率は次式で定義される。

$$\text{ConstraintRate} = \frac{N_{\text{success}}}{N} \tag{1}$$

3.2.2　補助指標（書き換えの性質・品質）

補助指標としては、書き換えの性質や品質を評価するために、語彙置換率（VRR）、語彙多様性（TTR）、n-gram 反復率、および実行時間（Time per Output）を用いる。

VRR は元文と生成文のトークン列を比較し、置き換わったトークンの割合を測定する指標である。値が高いほど大きな言い換えが行われていることを意味し、意味保持とのトレードオフを捉えるために用いる（RQ2）。ここで、X は元文のトークン列、X_diff はそのうち生成文と異なるトークンの集合とする。

$$\mathrm{VRR} = \frac{|X_{\mathrm{diff}}|}{|X|} \tag{2}$$

TTR は生成文におけるユニーク語彙の割合を測定し、過度な語彙の繰り返しや極端に限定された語彙への偏りがないかを確認する指標であり、こちらも RQ2 の検証に寄与する。Y は生成文のトークン列、types(Y)は Y に含まれるユニーク語彙の集合とする。

$$\mathrm{TTR} = \frac{|\mathrm{types}(Y)|}{|Y|} \tag{3}$$

n-gram 反復率は、同一の bi-gram や tri-gram がどの程度繰り返されているかを測定し、不自然な反復や定型表現への過度な依存を検出するために用いる。N_ngram は全 n-gram の数、N_ngram,rep は 2 回以上出現した n-gram の数とする。

$$\mathrm{RepRate}_n = \frac{N_{\mathrm{ngram,rep}}}{N_{\mathrm{ngram}}} \tag{4}$$

実行時間は、1 文あたりの処理時間を測ることで、逐次生成方式と一発生成方式の計算コストの違いを把握し、RQ1 を「実用性」の観点から評価するための指標となる。

3.3　比較対象となる生成方式

以上のタスク定義と評価指標に基づき、本研究では「一発生成」と「逐次生成」という二つの方式を比較する。一発生成方式（oneshot）は、元文と禁止文字集合をまとめて LLM への入力とし、単一の応答としてリポグラム文を生成する方式である。プロンプト設計のみで制約を伝える一般的な利用形態に近く、対話型 LLM をそのまま用いた素朴なベースラインとして位置づけられる。

これに対して逐次生成方式（sequential）は、元文を文や文節に分割し、禁止文字を含むトークンを検出したうえで、その部分のみを文脈を考慮しながら順番に書き換えていく方式である。各ステップで読みベース制約を強くかけ直し、必要に応じて再試行を行うことで、制約遵守率の向上を狙った「リポグラム専用」の強力な方式とみなせる。両方式を同一のデータセット・同一のモデル条件のもとで評価することで、日本語リポグラム生成においてどのようなトレードオフが存在するのかを明らかにすることが、本章で整理した研究課題の核心である。

第4章　提案する手法とシステム構成

本章では、日本語リポグラム生成のために構築したシステム全体の構成と、LLM を用いた生成手法の概要、さらにそれを利用者がブラウザから操作できるようにする Web アプリケーションの実装概要について述べる。

4.1　リポグラム生成フレームワークの概要

本研究で用いるリポグラム生成フレームワークは、入力文と禁止文字集合を受け取り、制約チェック・生成・再書き換えを組み合わせてリポグラム文を生成するパイプラインとして実装されている。中心的なコンポーネントは、文全体を書き換える一発生成方式（oneshot）と、文・トークン単位で逐次的に書き換える逐次生成方式（sequential）を切り替え可能なリライトモジュールであり、いずれの方式でもプロンプト内で禁止かなとリポグラムのルールを明示的に提示し、「説明なしで書き換え後の文のみを出力する」ように LLM に指示している。

制約チェックは、文字ベースと読みベースの二段階で行う。まず表記上の文字列に禁止かなが含まれるかを確認し、次に形態素解析で得られた読みをひらがなに正規化した上で、禁止かなが含まれるかを判定する。これにより、漢字表記の揺れをまたいだ検査が可能になる。生成結果が制約を満たさない場合には、再生成や局所的な修正を行うことで、制約遵守率の向上を図っている。

これら一連の処理の流れを図 1 に示す。図 1 では、入力文と禁止文字集合から始まり、前処理・制約チェックを経て、一発生成方式（oneshot）と逐次生成方式（sequential）の二つの経路に分岐し、再度の制約チェックを通じて最終的なリポグラム文を出力するまでのパイプライン構造を表している。

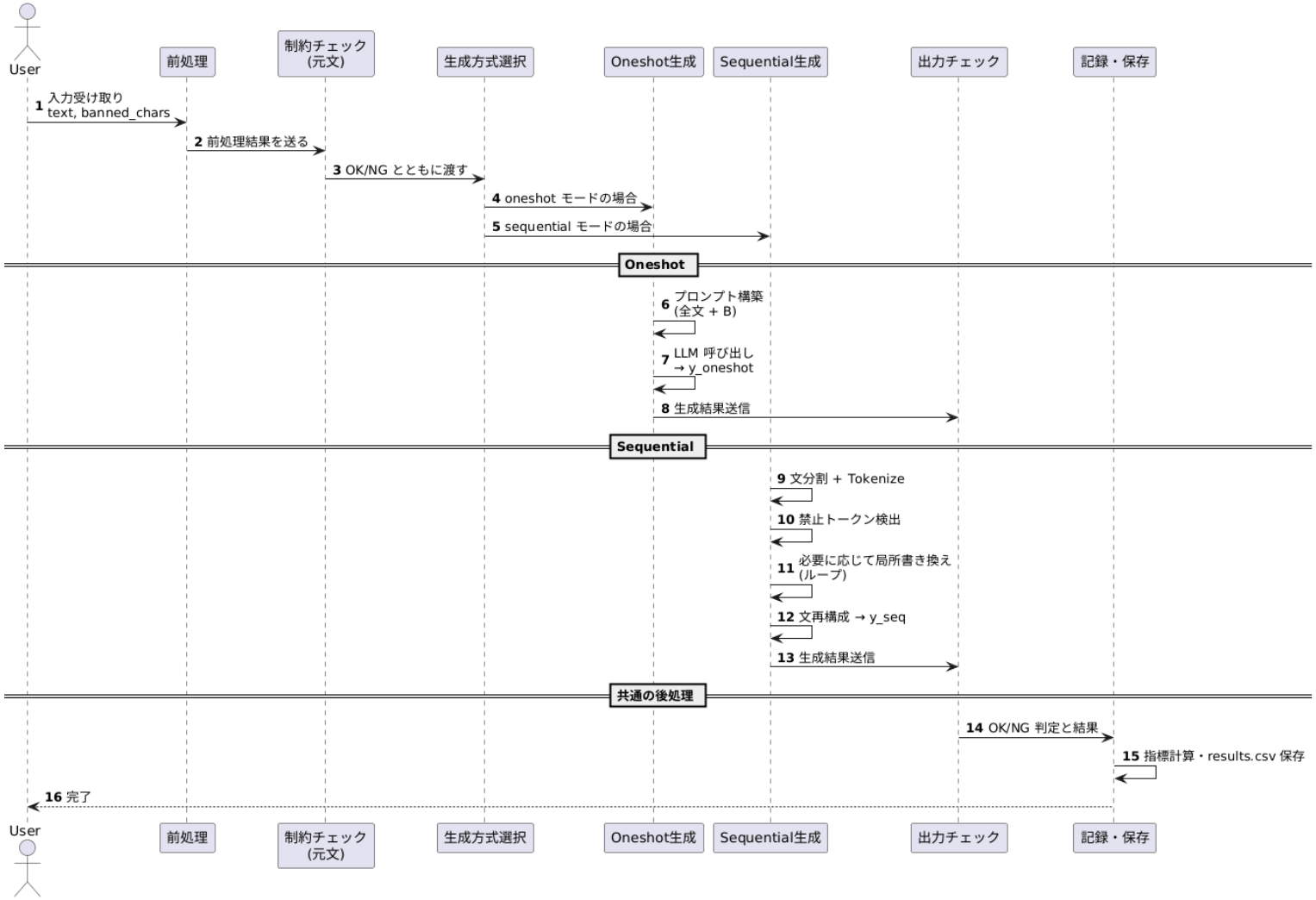

図 1 リポグラムの生成フロー

4.2　oneshot/sequential 方式

第 3 章では、oneshot と sequential という二つの生成方式の役割を概念的に整理した。本節では、その違いが実装上どのような設計の差として現れているかに焦点を当てる。両方式の主な違いをまとめたものを表 1 に示す。

表 1 生成方式の比較

項目	一発生成方式（oneshot）	逐次生成方式（sequential）
生成単位	文全体を一度に生成	禁止箇所を中心に逐次生成
再生成戦略	制約違反がある場合は原則として全文を再生成する	問題箇所のみを局所的に再生成し、必要に応じて複数回の書き換えを行う
プロンプトの文脈	元文全体と禁止文字集合のみをまとめて与える	元文全体に加え、対象トークン周辺の文脈（前後のトークン・文）も含める
API コール数	1 文あたりほぼ 1 回（高速）	禁止トークン数に比例して増加（遅め）
制御の粒度	文全体レベルで制御	トークン/文節レベルで細かく制御

一発生成方式（oneshot）では、元文と禁止文字集合をまとめて LLM に入力し、1 回の応答で全文をリポグラム文に書き換える。プロンプトには、禁止かなを読みレベルで一切含めないこと、原文の意味を大きく損なわないこと、日本語として自然な文とすることなどを明示し、出力形式を「書き換え後の文のみ」に限定する。制約違反が検出された場合には、全文を対象とした再生成を行う以外に細かな介入がしづらく、生成プロセスは基本的に「一回勝負」の設計となっている。

逐次生成方式（sequential）では、まず元文を文単位に分割し、各文を形態素解析によってトークン列に変換する。そのうえで、表記または読みが禁止かなを含むトークンを検出し、それらをターゲットとして局所的な書き換えを行う。

次に、検出された各禁止トークンについて、小さな書き換えループを回す。文全体や前後のトークンなどの文脈情報を含むプロンプトを組み立て、「この部分だけを禁止かなを含まない形に言い換える」ように LLM に指示し、その出力に対して制約チェックを行う。もし生成結果にまだ禁止かなが残っている場合は、プロンプトの条件を段階的に緩めながら再生成を行う。具体的には、元の語に近い同義語、意味的に近い上位概念、より自由度の高い意訳といった方向に少しずつ探索範囲を広げ、あらかじめ定めた最大試行回数の範囲で制約を満たす候補を探索する。このように sequential は、文全体を再生成するのではなく、問題箇所に対して段階的に介入する設計になっており、その分 API コール数と処理時間が増える一方で、制約遵守のための制御自由度が高いという特徴を持つ。

4.3　Web アプリケーション実装

リポグラム生成手法の検証と並行して、本研究ではブラウザ上から日本語リポグラムを生成・比較できる Web アプリケーションを開発した。図 2 に、その全体的な構成を示す。これは、研究者自身がコマンドラインからスクリプトを実行するだけでなく、非専門家や評価協力者が容易にリポグラムを体験できる環境を整備することを目的としている。また、将来的に主観評価実験（可読性・意味保持・自然さの評価）を行う際のインタフェースとしても利用することを想定している。

Web アプリの基本的な機能は、(1)入力文と禁止文字集合の指定、(2)生成方式（oneshot/sequential）の選択、(3)生成結果の表示と比較、の三つに整理できる。ユーザはブラウザ上のテキストボックスに元文を入力し、カンマ区切りで禁止ひらがなを指定することで、アプリ側に生成リクエストを送信する。裏側では、リポグラム生成フレームワークを呼び出して LLM による書き換えを行い、その結果を画面に返す。画面上では、元文・禁止文字・生成文が並べて表示されるため、どの文字が避けられているか、意味や文体がどのように変化したかを直感的に確認できる。

図 2 に示すように、アプリは大きく、ブラウザ上の UI、生成ロジックを担うアプリケーション部分、LLMAPI との連携およびログ記録を行う層の三つに分けて考えることができる。この構成により、一括評価用のコードとロジックを共有しつつ、インタラクティブな利用や将来の主観評価実験の実装を容易にしている。

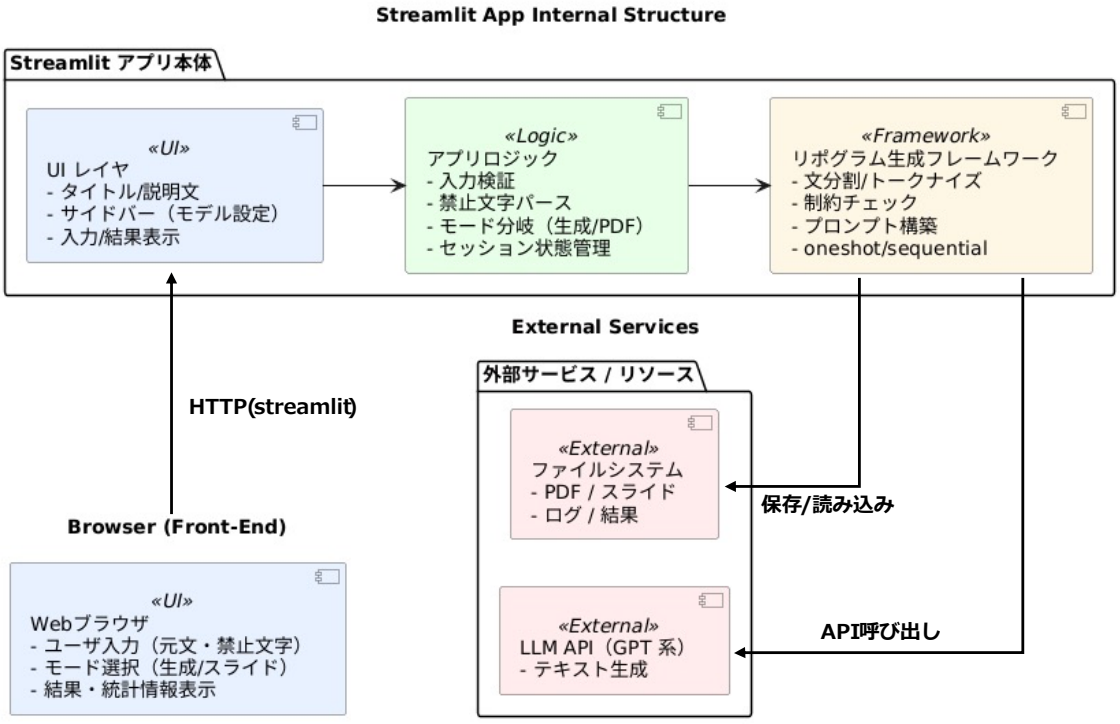

図 2 web アプリのシステム構成図

第5章　実験の設定と結果の概要

本章では、日本語リポグラム生成手法の評価に用いたデータセットと実験条件、および現時点で得られている定量的な結果とその解釈について述べる。

5.1　データセット

実験には、Tatoeba 由来の日本語例文コーパスを元に構築した制約付きデータセットを用いた。まず、文字数 20〜60 程度で、ひらがなまたは漢字を含む文を条件として抽出し、ランダムに 200 文を選択してベース文集合を作成した。その後、各ベース文に対して複数の禁止文字集合（banned_chars）を付与することで、制約付きパターンを展開し、最終的に 465 パターンからなる評価用データセットを構成した。

禁止文字集合は、単音禁止の easy 条件と行禁止の medium 条件に大別される。easy 条件では「い」「さ」「ら」など頻度の中程度の仮名を 1 文字だけ禁止し、medium 条件では「あ,い,う,え,お」や「か,き,く,け,こ」といった行単位の禁止を課した。各ベース文に対し、禁止セット内のいずれかのかなが表記に含まれる場合のみ、その組み合わせを 1 パターンとして採用し、1 文あたり最大 3 パターンまでに制限している。これにより、現実的な難易度の制約付き文集合を構成した。評価用データセットの一部を図 3 に示す。

	A	B	C	D	E	F
1	id	text	banned_chars	constraint_ty	genre	source_id
2	0	彼がこういうことを言ったのだと思われる。	い	easy	tatoeba	tatoeba:121021
3	1	彼がこういうことを言ったのだと思われる。	あ,い,う,え,お	medium	tatoeba	tatoeba:121021
4	2	彼がこういうことを言ったのだと思われる。	か,き,く,け,こ	medium	tatoeba	tatoeba:121021
5	3	木綿のミットをつければ赤ちゃんは自分の顔をひっかかなくなる。	か,き,く,け,こ	medium	tatoeba	tatoeba:80063
6	4	ビートルズの髪型はセンセーションを引き起こした。	か,き,く,け,こ	medium	tatoeba	tatoeba:197677
7	5	家にいるときはいろいろとやることが多くTVを見る暇もない。	い	easy	tatoeba	tatoeba:187123
8	6	家にいるときはいろいろとやることが多くTVを見る暇もない。	あ,い,う,え,お	medium	tatoeba	tatoeba:187123
9	7	家にいるときはいろいろとやることが多くTVを見る暇もない。	か,き,く,け,こ	medium	tatoeba	tatoeba:187123
10	8	君は約束を守るべきだと彼女は続けていった。	い	easy	tatoeba	tatoeba:176839

図 3 データセットのイメージ

5.2　実験条件と評価指標

モデルには OpenAI の GPT 系 LLM を用い、実験時点では gpt-4.1-nano を基準モデルとした。oneshot と sequential の双方について、温度やトップ確率などの生成パラメータは共通設定とし、各パターンに対して 1 回ずつ生成を行った。生成と評価は専用のバッチスクリプトにより自動化し、その結果を行ごとのレコードを持つ表形式データとして保存した。

評価指標としては、第 3 章で定義した読みベース制約遵守率を主指標としつつ、語彙置換率（VRR）、語彙多様性（TTR）、n-gram 反復率、実行時間を併用した。制約遵守率は、生成文の読み（カタカナをひらがなに正規化した文字列）に禁止かなが一切含まれない文の割

合として定義し、VRR は元文と生成文のトークン列の差分から「どれだけ単語が置き換わったか」を近似的に測る指標である。TTR と n-gram 反復率は生成文の多様性や反復傾向を捉える指標であり、実行時間は 1 文あたりの処理時間を通じて方式ごとの計算コストを評価するために用いた。主な実験条件と使用した指標の対応関係を表 2 にまとめる。

表 2 実験条件と評価指標

項目	設定
モデル	gpt-4.1-nano（OpenAI GPT 系 LLM）
生成パラメータ	温度や top-p などは両方式で共通設定
実行回数	各データパターンにつき各方式 1 回生成
主指標	読みベース制約遵守率
補助指標	語彙置換率（VRR）、語彙多様性（TTR）、n-gram 反復率、1 文あたり実行時間

5.3　制約遵守率と方式比較

まず RQ1「逐次生成方式は一発生成方式に比べて、どの程度制約遵守率を改善できるか」を検証するため、データセット全体に対する読みベース制約遵守率を比較した。結果を図 4 に示す。reading ベースの成功率は、sequential が 91.6%（426/465）、oneshot が 12.5%（58/465）であり、逐次生成方式が一発生成方式を大きく上回ることが分かる。

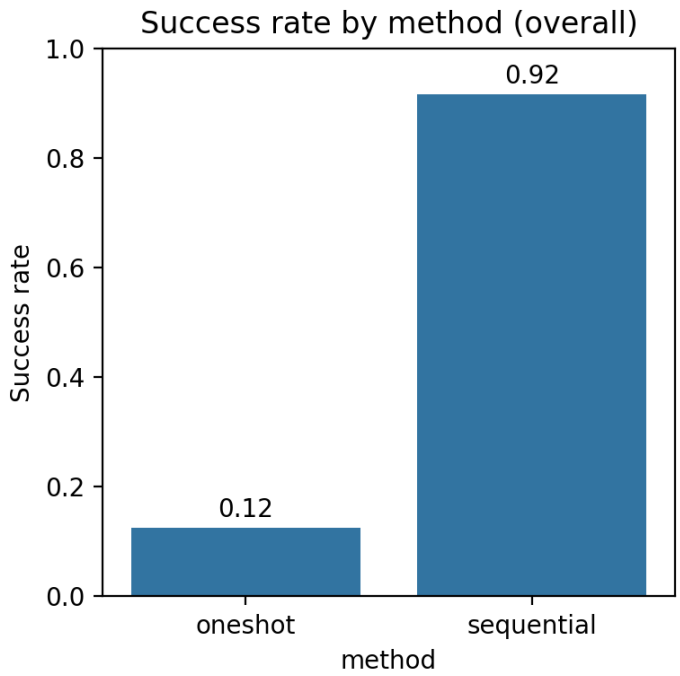

図 4 生成方式別の制約遵守率

さらに、制約タイプ別の成功率を図 5 に示す。とくに行禁止（medium）条件において、oneshot の成功率が大きく低下する一方で、sequential は高い成功率を維持しており、日本語の制約が厳しい条件ほど逐次生成方式の優位性が顕著になる傾向が見られた。また、各 id ごとに「どちらが成功したか」を比較したところ、sequential のみ成功したペアが 368（79.1%）、両方式とも成功または失敗の「同じ結果」が 97（20.9%）、oneshot のみが成功した例は 0 であった。対応のある t 検定でも、成功フラグ（0/1）の差は $t \approx 42, p \ll 0.001$ と、統計的に極めて有意な差が確認されている。

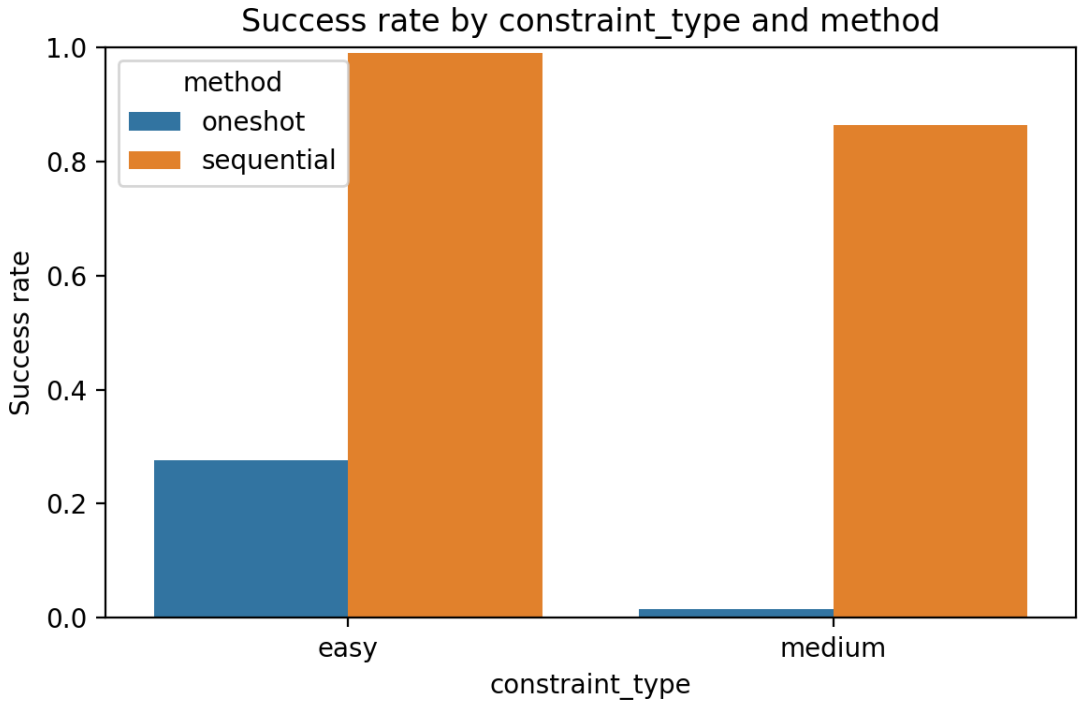

図 5 制約タイプ別×生成方式別の制約遵守率

5.4　VRR・TTR とエラー分析

次に、RQ2「制約遵守率を維持・向上させたうえで、元文の意味保持や自然さをどの程度保てるか」を検証するため、両方式とも制約を満たしたペア（58 件）のみを対象に、VRR と TTR を比較した。図 6 は、このときの VRR と TTR の分布を示している。VRR については、oneshot の値が sequential よりも有意に高く（平均差≈0.33, $p \ll 0.001$）、oneshot の方がより大きな言い換えを行う傾向があることが分かる。一方で、TTR には有意な差が見られず、語彙多様性そのものは両方式で大きくは変わらない。このことから、逐次生成方式は制約を守りながら原文に近い表現にとどまりやすく、oneshot はより大胆な言い換えを行いやすいという性質が示唆される。

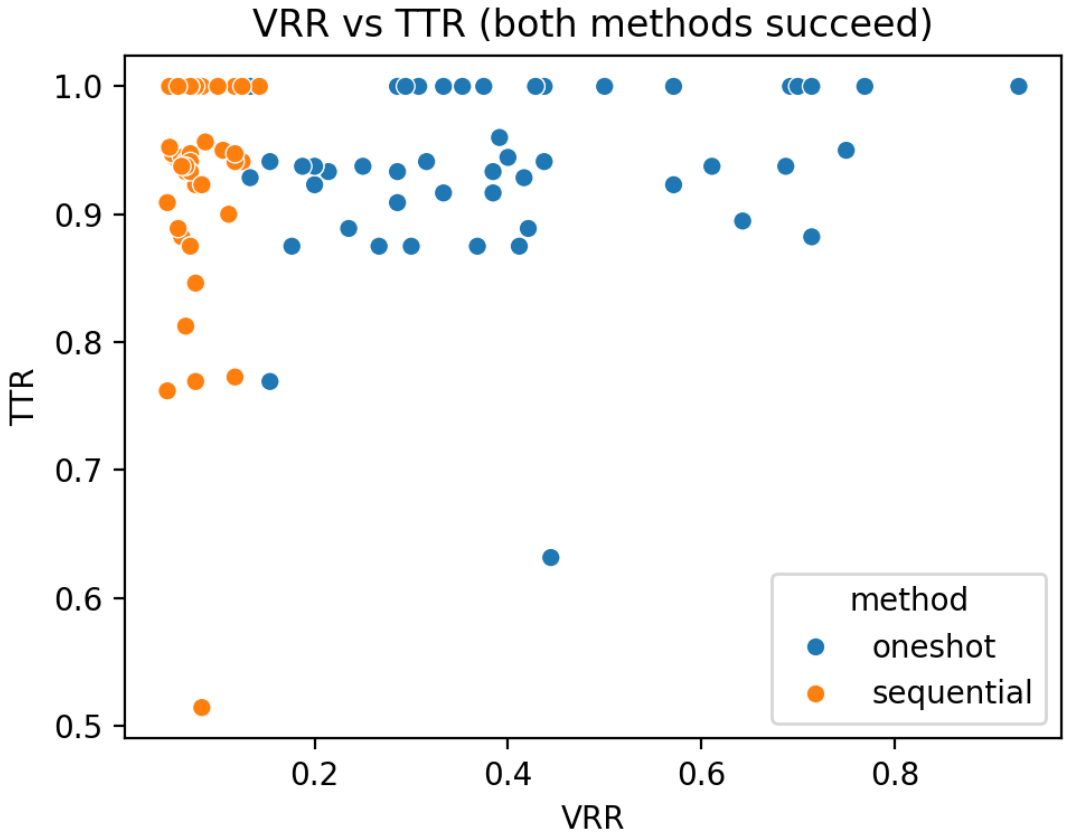

図 6 VRR と TTR の散布図

RQ3「制約タイプや文長・文ジャンルによって両方式の得意・不得意はどのように変化するか」については、エラー分析を通じて定性的に検討した。禁止文字が頻出助詞（「は」「が」「を」など）に含まれる場合や、行禁止によって多数の仮名が一度に制約対象となる場合、あるいは読みベース制約により漢字語の読みが制限される場合などでは、両方式とも意味保持が難しくなることが確認された。図 7 に、代表的な成功例・失敗例をいくつか示す。oneshot では文全体を丸ごと言い換える過程で意味が原文から大きく逸脱してしまう例があり、sequential では局所的な修正を重ねるうちに文全体の統一感がやや損なわれる例が見られた。これらの事例は、前章で述べた Web アプリケーション上で元文・制約・両方式の出力を並べて提示することで、今後の主観評価や改善策の検討に活用できる。

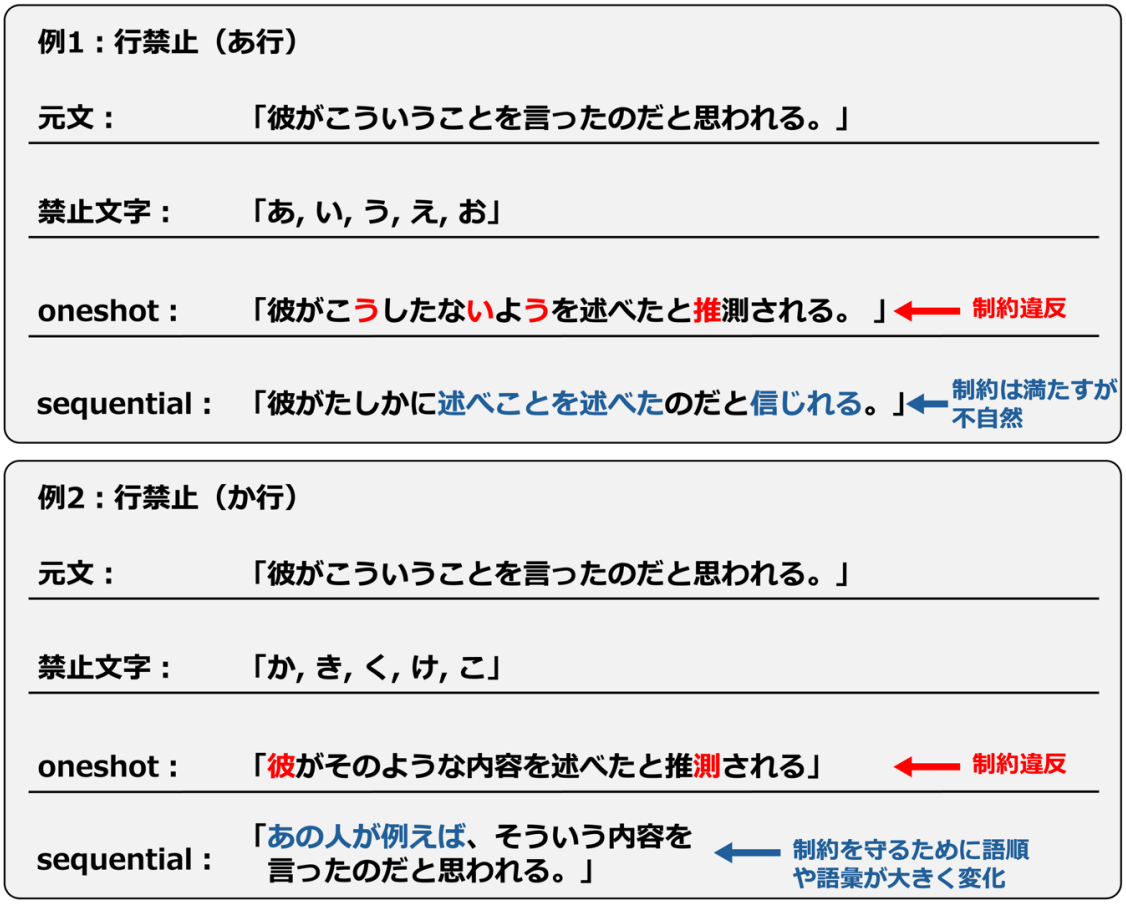

図 7 成功例・失敗例（元文／制約／各方式の出力）

第6章　今後の計画

本章では、これまでに構築したリポグラム生成フレームワークと評価環境を踏まえ、今後どのように研究を発展させていくかについて、実験拡張・主観評価・応用検証の観点から計画を整理する。

6.1　実験拡張の方針

現在の評価用データセットに基づく比較実験を出発点として、データセットと制約設定の拡張を行う。具体的には、次のような方向性を予定している。

6.1.1　制約タイプの拡張

現在は、単音禁止の easy 条件と行禁止の medium 条件を中心に検証を行っている。今後は、これに加えて、助詞を含む高頻度文字禁止や複数行の組み合わせなど、より現実的かつ難易度の高い制約条件を導入する。また、読みベース制約を本格的に適用し、表記の違いをまたいだ制約の影響を分析する。

6.1.2　文ジャンル・文長の多様化

ことわざや短文だけでなく、小説文、説明文、会話文など複数ジャンルを含む文集合を構築し、ジャンルごとに制約遵守率や意味保持の傾向を評価する。あわせて、文長の異なるサブセットを作成し、長文化が制約遵守および意味保持に与える影響を調べる。

6.1.3　反復実験と統計的検定の高度化

各条件について複数回の反復生成を行い、平均値と分散・信頼区間を算出したうえで、対応のある t 検定や混合効果モデルなどを用いて方式間の差を検証する。これにより、単一試行に依存しない頑健な比較結果を得ることを目指す。

6.2　主観評価と Web アプリケーションの活用

定量指標だけでは把握しきれない可読性・意味保持・自然さを補完するため、Web アプリケーションを評価インタフェースとして活用した主観評価実験を計画している。

6.2.1　評価プロトコルの設計

評価対象となる文について、oneshot と sequential の出力の順序をカウンタバランスした形で提示し、「どちらが読みやすいか」「どちらが元の意味をよく保っていると思うか」といった二者選択を行ってもらう。あわせて、可読性・意味保持・自然さなどを 7 段階評価など

で評定してもらい、自動指標（VRR、TTR、制約遵守率など）との相関を分析する。

6.2.2　Web アプリ側の機能拡張

既存の生成画面に、評価者が主観評価ラベルを入力できる簡易フォームを追加し、その結果を生成ログと一体で保存できるようにする。また、複数候補の一括提示や、評価済み・未評価の管理といった最低限の実験支援機能を実装することで、主観評価実験を効率的に実施できる環境を整える。

6.3　精度改善と応用領域の検証

これまでの実験により、逐次生成方式が制約遵守率の面で優位であることは確認できたが、意味保持や自然さの面では改善の余地が残っている。また、リポグラム生成は応用領域の観点からも潜在的な有用性を持つ。今後は、以下のような方向で精度改善と応用領域の検証を進める。

6.3.1　精度改善のためのアプローチ探索

精度改善に向けては、単に既存の oneshot/sequential 方式のパラメータを微調整するだけでなく、書き換えのフローや評価指標そのものを含めた複合的な改善が必要である。本項では、プロンプト設計・候補語選定・デコーディング制御・評価指標設計といった複数のレイヤにまたがるアプローチを整理し、どの部分をどのように拡張していくかの方向性をまとめる。具体的には、以下のような観点から段階的に検証を進める。

- プロンプト設計と再生成戦略の比較

  プロンプト設計や再生成戦略（再試行回数、候補選択ルールなど）を系統的に変更し、制約遵守率と意味保持の両立がどこまで可能かを検証する。プロンプト中での制約の提示方法や、失敗時に全文を再生成するか局所的に修正するかといった戦略を複数パターン用意し、本研究で用いている評価指標を通じて比較する。

- ハイブリッド型書き換えパイプラインの検討

  形態素解析や類義語辞書（例：WordNet や Bert など）を用いて、禁止文字を含まない候補語を事前に抽出し、その候補集合の中から LLM が文脈に最も適した語を選択するモデルのハイブリッド型の書き換えパイプラインを検討する。ルールベースのフィルタリングと LLM による文脈判断を組み合わせることで、探索空間を絞りつつ自然な言い換えを維持できるかを評価する。

- デコーディング段階での制約制御の導入

  デコーディング段階での制約制御として、負の語彙制約（negative lexical constraints）や

ログ確率のバイアス付与などを利用し、生成途中で禁止トークンが選択されにくくなるような仕組みを比較検討する。従来のプロンプトベースの制約提示のみの場合と比較し、制約違反の頻度や生成の安定性がどの程度改善するかを確認する。

- 重み付き指標による書き換えの質的評価

  重要語に重みを付与した重み付き語彙制限率（Weighted Vocabulary Restriction Rate; WVRR）などの指標を導入し、「どの語をどの程度置き換えたか」をより精緻に評価しながら、意味保持と制約遵守のトレードオフを分析する。特に、機能語やキーワードが書き換えられた場合の影響を重みづけして扱うことで、単純な置換率では捉えきれない質的な変化を定量的に把握することを目指す。

- 主観評価フィードバックに基づくチューニング

  主観評価の結果をフィードバックとして用い、人間の評価と自動指標のギャップを縮めるチューニング方針を検討する。例えば、「可読性は高いが VRR が高すぎるケース」や「制約は守られているが意味が損なわれているケース」を抽出し、その特徴を分析したうえでプロンプトや再生成戦略に反映させることで、人間の感覚により近い評価軸にモデルを調整していく。

6.3.2 応用領域の検証

リポグラム生成の技術は、純粋な制約付き文章生成の枠を超えて、創作支援・出力制御・コミュニケーション支援など複数の応用可能性を持つ。ここでは、今後具体的に検証していきたい応用領域を整理し、それぞれについて実現像と評価の方向性を示す。

- 文芸作品への応用

  リポグラム小説や詩の自動生成を支援するツールとして用い、作家や学生との協働実験を通じて、創作表現の幅をどの程度広げられるかを検証する。その際には、生成された作品の完成度だけでなく、「制約が着想や表現をどのように変化させたか」といった創作プロセスへの影響もインタビューやワークショップ形式で評価することを想定している。

- 生成モデルの出力制御への応用

  大規模言語モデルの出力から NG 語やセンシティブな表現を含まない自然な言い換えを生成する仕組みとして組み込み、既存の単純な出力フィルタよりも文脈を保った制御が可能かを検討する。具体的には、ニュース要約や対話応答などのタスクに対して、禁止語リストを与えたときの可読性・意味保持・残余リスクを比較する実験デザインを想定している。

- 吃音や発話困難者の支援

  言いにくい音や語を含まない自然な言い換え表現を提示することで、発話負荷の軽減やコミュニケーション支援にどの程度貢献できるかを、専門家との協議や小規模な利用実験を通じて探る。まずは、音声言語聴覚の専門家と協力して「避けたい音・語」の例を整理し、本研究のリポグラム制約にどのようにマッピングできるかを検討したうえで、プロトタイプ段階の評価を行うことを目標とする。

第7章　まとめ・期待される成果

7.1　本研究の現時点でのまとめ

本研究は，日本語におけるリポグラム生成を対象として，LLM を用いた二つの方式（oneshot と sequential）を比較しつつ，制約遵守・意味保持・文法性をバランスよく満たす生成手法を検討することを目的としている。中間報告の段階までに，Tatoeba 由来の日本語例文から構築した評価用データセットを用いた評価パイプラインを整備し，実際に両方式を同一条件で比較する実験を行った。その結果, 読みベース制約遵守率に関しては sequential が oneshot を大きく上回り，特に行禁止（medium）条件において顕著な差があることが確認された。一方，両方式とも成功したケースに限れば，oneshot はより大きな言い換え（高い VRR）を行うが，語彙多様性（TTR）自体は両方式で大きく変わらないことも分かった。

また, リポグラム生成フレームワークの実装と並行して, ブラウザ上から元文・禁止文字・生成方式を指定してリポグラムを生成・比較できる Web アプリケーションも構築した。これにより，コマンドラインからの一括評価だけでなく，個別の文に対して逐次的に試行し，失敗例や興味深い生成例を簡便に収集できる環境が整った。Web アプリケーションは，今後予定している主観評価実験のインタフェースとしても活用可能であり，「手法」と「評価環境」の両面で基盤が整いつつあるといえる。

7.2　期待される成果と意義

学術的な観点からは，日本語リポグラム生成というこれまでほとんど扱われてこなかったタスクに対して，明確なタスク定義・評価指標・実験プロトコルを提示し，LLM を用いた具体的な生成方式を比較した点に意義がある。特に，文字レベルだけでなく読みベースの制約を導入した評価や，逐次生成方式のような専用フレームワークが oneshot ベースラインに対してどの程度優位になりうるかを定量的に示したことは，今後の制約付き生成研究にとって有用な知見となることが期待される。

応用的な観点からは，リポグラムの枠を超えて，NG ワードを避けながら意味を保つ応答生成や, 発音困難な音を含まない言い換え表現の生成, 言語学習における制約付き作文課題の自動生成など，さまざまな応用シナリオが考えられる。本研究で得られた「どの程度まで制約を強くしても自然な文が保てるのか」「逐次的な書き換えはどのような条件で有効か」といった知見は，これらの応用タスクにも転用可能である。

7.3　今後の課題

一方で，本研究にはいくつかの課題も残されている。第一に，現時点の実験はデータセット規模や制約タイプが限定的であり，文ジャンルや文長, 制約の種類を拡張しても同様の傾

向が成り立つかどうかは今後の検証が必要である。第二に，評価は主として自動指標に依拠しており，可読性・意味保持・自然さに関する主観評価との対応関係はまだ十分に明らかになっていない。第三に，逐次生成方式は制約遵守率の面で優れている一方で，計算コストや実装の複雑さといった点から，実用性の観点での検討も必要である。

卒業研究としては，これらの課題に対して，データセットと制約設定の拡張，主観評価実験の実施，さらには sequential と oneshot を組み合わせたハイブリッド方式の検討などを通じて，日本語リポグラム生成におけるより汎用的な設計指針を示すことを目標とする。

参考文献

Kajiwara T.，Negative Lexically Constrained Decoding for Paraphrase Generation，Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics（ACL 2019），pp. 6047–6052，2019 年.

Loujain L.，lipogram_e: French lipogram text generation with GPT models，GitHub リポジトリ，URL: https://github.com/Loujainl/lipogram_e，最終アクセス日: 2025 年 11 月 24 日.

Shen T., Lei T., Barzilay R., Jaakkola T., Style Transfer from Non-Parallel Text by Cross-Alignment, Advances in Neural Information Processing Systems 30（NeurIPS 2017），2017 年.

Li J.，Jia R.，He H.，Liang P.，Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer，Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies（NAACL-HLT 2018），pp. 1865–1874，2018 年.