一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

リポグラム生成プログラムの開発および評価分析

橋本 葵† 飯尾 淳†

†中央大学国際情報学部 〒162-0843 東京都新宿区市谷田町 1-18
E-mail: †a22.fr7w@g.chuo-u.ac.jp

あらまし 本研究は，日本語におけるリポグラム自動生成を対象とし，特定の禁止文字を含まない文章を生成する手法を提案する．形態素解析と制約チェックに基づき，大規模言語モデルを用いた一発生成方式と逐次生成方式を設計し，制約遵守率，意味保持，文法性，多様性，計算コストの観点から比較評価する．実験の結果，日本語特有の表記体系や助詞制約が性能に与える影響を明らかにした．

キーワード リポグラム，自然言語処理，LLM

Development and Evaluation of a Lipogram Generation Program

Aoi HASHIMOTO† Jun IIO†

†Faculty of Global Informatics, Chuo University 1-18 Ichigayatamati, Shinjuku-ku, Tokyo, 162-0843 Japan
E-mail: †a22.fr7w@g.chuo-u.ac.jp

Abstract This study focuses on the automatic generation of Japanese lipograms and proposes a method for producing texts that exclude specified prohibited characters. Based on morphological analysis and constraint checking, we design two large language model–based approaches: a oneshot approach and a sequential approach. We evaluate these approaches in terms of constraint satisfaction, semantic preservation, fluency, diversity, and computational cost. Experimental results reveal how characteristics unique to the Japanese writing system and particle-related constraints affect generation performance.

Keywords lipogram, natural language processing, large language models (LLM)

1. はじめに

本章では，日本語におけるリポグラム自動生成という課題の背景と,本研究の目的･位置づけを整理する．とくに，細かい文字レベルの制約の下で自然な文章を生成することの難しさと，それに対して本研究がどのようなアプローチをとるのかを明らかにする．

1.1. 研究背景

近年,大規模言語モデル（Large Language Model; LLM）の発展により，日本語を含む多言語において自然な文を自動生成することが容易になりつつある．敬語変換やポジティブ表現への言い換え，NG ワードを含まない応答生成など，さまざまな制約付き文章生成タスクへの応用も進んでいる．とくに，Web サービスや対話システムでは，有害表現や差別的表現を含まない出力を求めるニーズが高く，出力制御の重要性は今後さらに増すと考えられる．

しかし，LLM に対して「この語を使わないでください」「この表現を避けてください」といった指示を与えても，禁止語や禁止文字を完全に排除することは難しい．単純な bad words フィルタによって出力後に文字列を置換する方法もあるが，その場合は文法性の破綻や意味の歪みが生じやすい．すなわち,「制約遵守」「意味保持」「文法性（自然さ）」を同時に満たす制約付き文章生成は，依然として難しい課題である．

1.2. 日本語リポグラムという題材

本研究では，このような制約付き生成の一例として,日本語リポグラムを題材に取り上げる．リポグラムとは，特定の文字を一切用いないという制約のもとで書かれた文章や作品を指し，英語圏では「e」を使わない小説などの著名な例が知られている．言語遊戯・実験文学としての側面が強い一方で,「細かい制約の下で意味の通る文章を書く」という点で，制約付き生成の良いテストベッドでもある．

日本語で同様の制約を課すと，アルファベット言語とは異なる難しさが現れる．助詞や活用語尾など高頻度の仮名を禁止すると，そもそも自然な文を構成すること自体が困難になる．また，漢字・ひらがな・カタカナが混在し，同じ読みを複数の表記で書けるという特性のため，表記レベルと読みレベルのどちらで制約を課すかによって，生成可能な文の範囲が大きく変化

する.さらに,日本語は空白による単語区切りがなく，単純な文字列操作だけでは意味を保った書き換えが行いにくい．このような理由から，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，自動生成はなおさら難しい課題となっている．

1.3. 本研究の目的

以上の背景を踏まえ，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）を自動生成する」手法を検討し，その特性を明らかにすることを目的とする．具体的には，文全体を一度に言い換える一発生成方式（oneshot）と，禁止文字を含む部分を検出して局所的に書き換える逐次生成方式（sequential）の二つの方式を用意し，同一のデータセット上で比較する．

評価の観点としては，(1)禁止文字を本当に 0 にできているかという制約遵守，(2)元の文の意味をどの程度保てているかという意味保持，(3)日本語として自然に読めるかという文法性・自然さ，に加え，(4)実行時間や再生成回数といった実用上のコストを重視する．本研究は，これらの観点に基づいて oneshot と sequential の振る舞いを比較し，日本語リポグラム生成における実用的な設計指針を得ることを目標とする．

2. 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・スタイル変換の研究動向を概観したうえで，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整理する．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踏まえ，本研究のアプローチとの違いを明らかにする．

2.1. 制約付き文章生成と NG ワードフィルタ

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，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，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て再生成する枠組みなどが提案されている．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」を使わない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，フランス語の「e」抜き小説を対象に GPT 系モデルを用いてリポグラム文を生成するプロジェクトも報告されている．

リポグラム作品としては，Perec（1969）のような著名な事例も知られており，制約下での創作という観点から広く参照されてきた．

しかし，これらは文字と音の対応が比較的単純なアルファベット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,形態素境界も明示的には現れない.そのため，表記だけを見て禁止文字を処理する単純なアルゴリズムでは，意味を保ったままリポグラム文を生成することは難しい．

日本語におけるリポグラム研究や実装例は，現時点で体系的な報告が多いとは言えない．本研究は，日本語という表記と読みが複雑に絡む言語において，禁止文字（禁止かな）を完全に回避する生成を扱う点に特徴がある．

2.3. 制約付き生成における本研究の位置づけ

LLM に対する制約付与の観点からは，negative constraints や制約付き生成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生成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．

なお，本研究の英語側の実装では，候補語の生成・選別に BERT（Devlin ら, 2019）と WordNet（Miller, 1995）を用いる構成を採用している．

本研究は，上記の観点を踏まえつつ，日本語リポグラムという難度の高い制約付き生成条件に対して，oneshot と sequential の二方式を比較し，どのような設計が現実的かを探る応用的な研究として位置づけられる．

3. 課題設定と評価指標

本章では，日本語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課題と評価指標を整理する．あわせて，本研究で用いるデータセットと制約タイプの概要も示す．

3.1. タスク定義

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x（ことわざや例文など）と，禁止する仮名の集合Bを与える．禁止文字は，ひらがな 1 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．

yは，表記上あるいは読みレベルで禁止文字を一切含まないことが求められる．同時に，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，元文の主要な意味内容を大きく損なっていないことが望ましい．より形式的には，文字ベース制約では「yに現れるどの文字も禁止集合Bに属さない」ことを，読みベース制約では「yを形態素解析して得られる読みの仮名列に現れるどの仮名もBに属さない」ことを条件として課す．

3.2. データセット概要

実験には，Tatoeba 由来の日本語例文コーパスなどを元に構築した制約付きデータセットを用いる．まず，文字数 20〜60 字程度で，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Bを付与することで，制約付きパターンを展開し，最終的に約 500 パターンからなる評価用データセットを構成した．

表 1 生成方式の比較

項目	oneshot	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	しない	問題箇所のみを局所的に再生成
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

表 1 では，ベース文集合（約 200 文）と，そこから展開した評価用データセット（約 500 パターン）について，文数・平均文長・制約タイプの内訳などの統計情報を整理する．禁止文字セットと制約タイプ

3.3. 禁止文字セットと制約タイプ

禁止文字集合Bは，難易度の異なる複数の制約タイプから構成される．easy 条件では，「い」「さ」など頻度の中程度のひらがなを 1 文字だけ禁止する単音禁止を用いる．これは，語彙選択の自由度が比較的高く，リポグラムとしては実現しやすい制約である．

medium 条件では，「あ,い,う,え,お」といった行単位の禁止を課す．行制約は同時に多数の仮名を禁止するため，使用可能な語彙が大きく制限され，自然な文を生成することが難しくなる．また，子音系列禁止のように，特定の子音に対応する仮名をまとめて禁止するパターンも含めて検証対象とする．

これらの制約タイプは，単音禁止から行禁止へと段階的に難易度が上がるように設計しており，どの程度の制約で oneshot と sequential の性能差が顕在化するかを観察することができる．評価指標

上記のタスクとデータセットに基づき，本研究では以下の指標を用いて生成結果を評価する．

1. 制約遵守率（主指標）

   生成文が禁止文字を一切含まない割合であり，リポグラムとして成立しているかどうかを判定する最重要の指標である．読みベース制約を採用する場合は，形態素解析と正規化を経て得られた読み列に対して判定を行う．

2. 語彙置換率（Vocabulary Replacement Rate; VRR）
   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．

3. 語彙多様性や反復の指標（Type-Token Ratio; TTR・n-gram 反復率など）
   TTR は「異なり語数÷総語数」で定義され，語彙の多様さを測る指標である．n-gram 反復率は，同じ 2 語列や 3 語列（bi-gram/tri-gram）がどの程度繰り返されているかを測定し，不自然な反復や語彙の偏りを検出するために用いる．

4. 実行時間・反復回数
   1 文あたりの処理時間や，sequential における書き換えループの回数を記録し，制約遵守率とのトレードオフを評価する．

これらの指標を組み合わせることで，「制約をどれだけ厳格に守れるか」「元文の意味と自然さをどこまで保てるか」「実用時間内で動かせるか」という観点から，oneshot と sequential の二方式を多面的に比較する．

4. 提案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フレームワークの全体像と，oneshot/sequential の具体的な手法，およびそれらをブラウザから利用可能にする Web アプリケーションの構成について述べる．

4.1. 全体フレームワーク概要

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，制約チェックと書き換え処理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，文やトークン単位で逐次的に書き換える逐次生成方式（sequential）であり，同じフレームワーク内で切り替えられるようになっている．

sequential 方式では，まず元の文を文単位・トークン単位に分割し，それぞれの表記と読みが禁止文字を含んでいるかどうかをチェックする．禁止文字を含むトークンが見つかった場合に限って，そのトークンを対象に LLM による言い換えを行い，置き換え候補が禁止文字を含まないかどうかを再度チェックする，という処理を繰り返す．

一方，oneshot 方式は元文全体を一度に LLM に入力して書き換えた後で，その結果に対して制約チェックを行う方式であり，事前のトークン単位の違反検出は行わない．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」ように指示している点は共通である．

4.2. oneshot 手法

一発生成方式（oneshot）では，元文と禁止文字集合 B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，「指定された禁止かなを読みレベルで一切含めないこと」「元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．

この方式の長所は，処理が単純で高速であり，実装も比較的容易な点にある．一方で，禁止文字が 1〜2 文字だけ残ってしまうケースが多く，とくに行禁止のような厳しい制約下では制約遵守率が低くなる傾向がある．また，生成過程に細かく介入しづらいため，失敗した場合には全文の再生成に頼らざるを得ない．

4.3. sequential 手法

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，各トークンの表記と読みを調べ，どちらか一方でも禁止かなを含むトークンを検出する．

禁止トークンが見つかった場合には，そのトークンをターゲットとして LLM による言い換えを行う．このとき，対象トークンを含む文（current_sentence）と元の文全体（full_text）を文脈としてプロンプトに含めることで，文脈に沿った自然な代替表現を生成できるようにする．生成された候補語についても，表記と読みが禁止かなを含まないかを再度チェックし，条件を満たさない場合は失敗履歴を更新しながら，あらかじめ定めた回数の範囲で再試行を行う．

このように sequential は，文全体をまとめて再生成するのではなく，「禁止文字を含むトークンが見つかった箇所にだけ局所的に介入する」設計になっている．そのため API コール数と処理時間は増える一方で，必要な箇所に集中的に書き換えを行えることから，oneshot より高い制約遵守率が期待できる．

4.4. 制約チェックの仕組み

制約チェックでは，各トークンについて「表記」と「読み」の両方に禁止かなが含まれているかを調べる．表記については，表層文字列に禁止かなが含まれていないかを単純に確認し，読みについては，形態素解析器から得られたカナ読みをひらがなに正規化したうえで，同様に禁止かなの有無を判定する．いずれか一方にでも禁止かなが含まれていれば，そのトークンは制約違反とみなされ，sequential 手法の書き換え対象となる．

このように，表記ベースと読みベースの双方で制約を確認することで，漢字・カタカナ・ひらがなの表記

揺れをまたいだ検査が可能になり,「表記を変えても読みとしては同じ禁止音を含んでいる」といったケースも検出できる.

4.5. Web アプリケーションのシステム構成

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．これは，コマンドラインからスクリプトを実行するだけでなく，評価協力者など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．

システム構成は，ブラウザ（フロントエンド），生成ロジックを担うアプリケーション層，LLM API との連携およびログ記録を行う層の三つに大別できる．ユーザはブラウザ上のフォームに元文と禁止ひらがな集合を入力し，oneshot/sequential のどちらか，あるいは両方を選択して生成を実行する．サーバ側では，第 4.1 節で述べたフレームワークを呼び出し，生成文と制約遵守状況を計算してから結果を返す．これらコンポーネント間のデータフローを図 1 に示す．

画面上では，元文・禁止文字・各方式の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．また，入力・出力・制約・モデル設定をログとして保存することで，後からの定量分析やエラー分析にも活用できる．将来的には，この Web アプリを主観評価実験のインタフェースとして用い，oneshot と sequential の出力を並べて提示し，可読性や意味保持について人手評価を収集することを計画している．

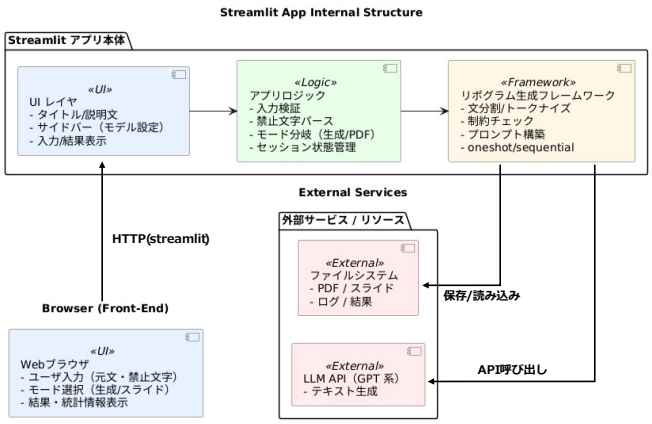


図 1 web アプリのシステム構成図

5. 実験設定と結果

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. データセットと実験条件

実験には，第 3 章 3.2 節および 3.3 節で定義した制約付きデータセットをそのまま用いた．

モデルには OpenAI の GPT 系 LLM（gpt-4.1-nano）を用い，実験時点ではこの 1 種類のモデルを基準とした．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．評価用データセットの一部を図 2 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 2 データセットのイメージ

5.2. 制約遵守率と方式比較

まず，読みベース制約遵守率を主指標として，oneshot と sequential を比較した．全体で見ると，sequential が oneshot を大きく上回ることが明らかになった．具体的な数値はここでは省略するが，イメージとしては sequential が 9 割前後の成功率を示す一方，oneshot は 1 割前後にとどまる，といった大きな差である．全体の成功率の比較は図 3 に示す．

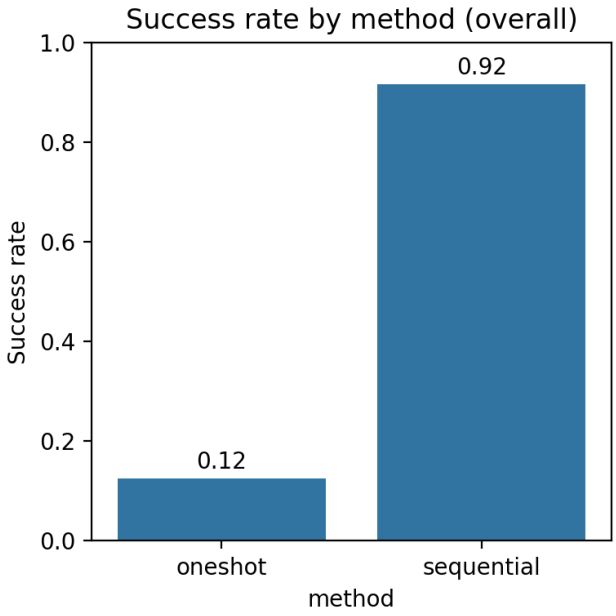


図 3 生成方式別の制約遵守率

各 id ごとに「どちらが成功したか」を比較すると，sequential のみ成功したペアが大多数を占め，両方式とも成功または失敗となる「同じ結果」のペアが残りを

構成する．oneshot のみが成功した例は一件もなく，対応のある t 検定でも，統計的に有意な差が確認された．とくに行禁止（medium）条件で差が顕著であり，厳しい制約下では sequential の優位性が際立つ結果となった．制約タイプごとの成功率の違いは図 4 に示す．

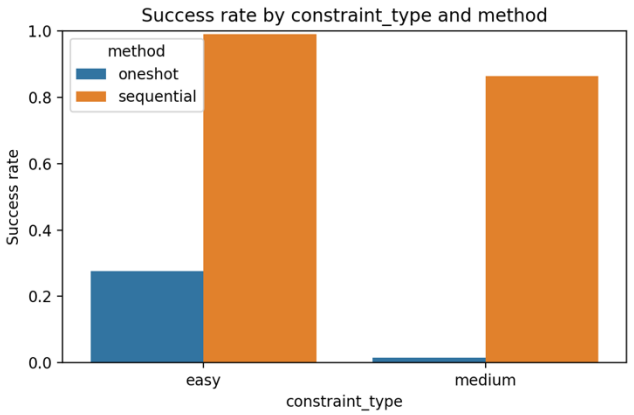


図 4 制約タイプ×生成方式別の制約遵守率

5.3. 補助指標と実行時間

次に，制約を満たしたケースに限定して，語彙置換率（VRR）や語彙多様性（TTR），実行時間などの補助指標を比較した．両方式とも制約を満たしたペアに絞って VRR を比較すると，oneshot の VRR が sequential より有意に高く，より大きな言い換えを行う傾向があることが分かった．一方，TTR については両方式で大きな差は見られず，語彙多様性そのものは同程度であることが示唆された．

また，VRR と制約遵守の関係を見るために，VRR の値と成功・失敗フラグの関係をプロットした結果を図 5 に示す．VRR が高いほど制約を守るのが難しくなる傾向があり，「大きく言い換えるほど失敗しやすい」という直感に対応する結果となっている．

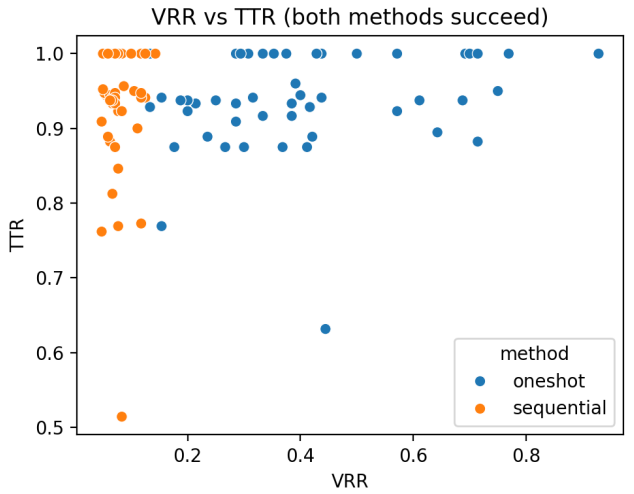


図 5 VRR と TTR の散布図

これらを総合すると，「制約を守りながら元文に近い表現を保つ」という観点では sequential が優位であり，「大胆に言い換えるが制約違反も起こしやすい」のが oneshot という性質が浮かび上がる．実行時間については，sequential が oneshot より長くなるのは当然であるが，制約タイプや文長によって増加幅がどの程度変化するかを確認したところ，medium 条件や長文ほどコストが増大する傾向が見られた．

5.4. エラー分析

定量的な指標に加えて，代表例を用いたエラー分析も行った．oneshot 手法の典型的な失敗パターンとしては，禁止文字が 1〜2 文字だけ残るケースや，助詞や頻出語に禁止文字が含まれているにもかかわらず十分に書き換えられていないケースが多く見られた．また，行禁止条件では，使用可能な語彙が大きく制限される結果として，意味が原文から大きく逸脱する例も確認された．

sequential 手法でも，制約が極端に厳しい場合や，読みベース制約によって漢字語の読みが強く制限される場合などには，再生成を繰り返しても自然な表現にたどり着けないケースが存在した．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もあり，逐次生成ならではの課題も浮かび上がっている．

図 6 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，第 4 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．

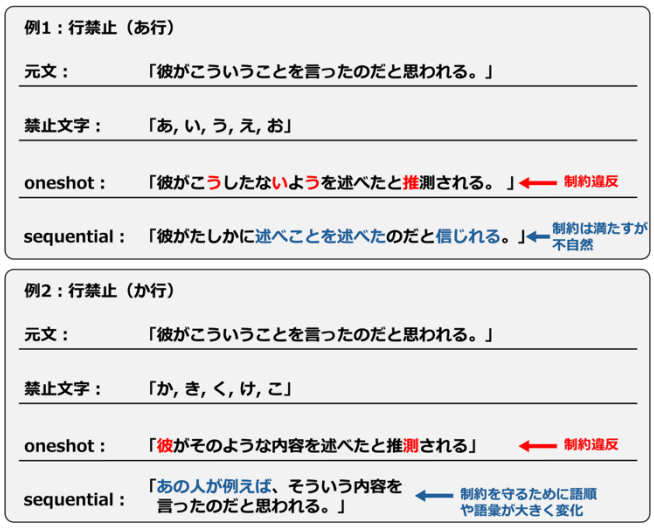


図 6 成功例・失敗例

6. まとめと今後の課題

本章では，本研究の内容を簡潔にまとめるとともに，

今後の課題と展望について述べる.

6.1. 本研究のまとめ

本研究は，日本語におけるリポグラム自動生成を題材として,「細かい文字制約の下で自然な文章を生成する」という制約付き文章生成の問題に取り組んだ．具体的には，特定の仮名を一切含まない日本語文を生成するタスクを定義し，単音禁止や行禁止といった複数の制約タイプを含む評価用データセットを構築した．

手法面では，文全体を一度に言い換える一発生成方式（oneshot）と，禁止文字を含む部分を検出して逐次的に書き換える方式（sequential）からなる生成フレームワークを設計した．さらに，読みベースの制約チェックを組み込むことで，漢字・かな表記の違いをまたいだ禁止音の検出を可能にした．また，これらの手法をブラウザから利用できる Web アプリケーションとして実装し，生成・比較・ログ取得を一体化した評価環境を構築した.

実験の結果，sequential は oneshot に比べて読みベース制約遵守率で大きく優れ，とくに行禁止など難度の高い制約下で顕著な差が確認された．制約遵守率と実行時間を比較すると, sequential は高い成功率と引き換えに一定の時間コストを要することが分かる.総じて，日本語リポグラム生成においては，逐次的な書き換えフレームワークが有効であることが示唆された.

6.2. 得られた知見

本研究を通じて得られた知見を，制約タイプ・書き換えの性質･ツールとしての側面の三点から整理する.

第一に，制約タイプごとの傾向として，easy な単音禁止条件では oneshot も一定程度制約を守ることができるが，medium の行禁止条件では sequential の優位性が明確になることが分かった．行禁止では使用可能な語彙が大きく制限されるため，全文を一度に生成する oneshot では微妙な制約違反を防ぎきれない一方で，sequential は局所的に試行を重ねることで成功率を高められる.

第二に，書き換えの性質に関しては，制約を守ったケースに限定すると, oneshot の方が語彙置換率（VRR）が高く，より大胆な言い換えを行う傾向がある一方，sequential は元文に近い表現を保ちつつ禁止文字を取り除く傾向があることが示された.語彙多様性（TTR）には大きな差は見られず，両方式とも極端に単調な表現にはなっていないことが確認された.

第三に，Web アプリケーションとしての側面では，生成過程や失敗例を可視化できることで，エラー分析や教育的な利用に有用であることが分かった．ユーザが自分で制約を設定して試せるデモ環境としても機能し，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る.

6.3. 今後の課題と展望

今後の課題としては，手法の拡張・評価の高度化・応用領域の検証の三点が挙げられる.

手法の拡張という観点では，助詞や高頻度文字を含む，より厳しい制約への対応が重要である．現状の sequential フレームワークをベースにしつつ, 形態素解析や類義語辞書を用いたハイブリッド型の書き換え，デコーディング段階での制約制御などを組み合わせることで，制約遵守と意味保持の両立をさらに改善できる可能性がある.

評価の高度化としては，VRR や制約遵守率といった自動指標に加え，人手による主観評価（可読性・意味保持・自然さ・面白さなど）を取り入れることが重要である. 本研究で構築した Web アプリを評価インタフェースとして用い，oneshot と sequential の出力を並べて提示し，どちらが好ましいかを評価してもらう実験を計画している.

応用領域の検証としては，NG ワード回避やセンシティブ表現の置換といった出力制御への応用，発話困難者が言いにくい音を避けるための言い換え支援，創作支援ツールとしてのリポグラム生成など，さまざまな展開が考えられる.本研究で得られた知見を踏まえ，これらの応用シナリオにおける実現可能性と課題を検証し，日本語リポグラム生成における実践的な設計指針を卒業研究としてまとめることが今後の目標である.

文　献

[1] C. Hokamp and Q. Liu, “Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search,” Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.1535-1546, Vancouver, Canada, July 2017.

[2] M. Post and D. Vilar, “Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation,” Proc. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp.1314-1324, New Orleans, Louisiana, June 2018.

[3] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying Conditional Random Fields to Japanese Morphological Analysis,” Proc. 2004 Conference on Empirical Methods in Natural Language Processing, pp.230-237, Barcelona, Spain, July 2004.

[4] J. Tiedemann, “The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT,” Proc. Fifth Conference on Machine Translation, pp.1174-1182, Online, November 2020.

付　　録

項目	値
制約付きパターン数（総数）	465
評価対象ユニーク文数（元文）	195
制約タイプ：単音禁止（easy）	195
制約タイプ：行禁止（medium）	270

付録 1 評価データセットの内訳

手法	成功数	総数	成功率
oneshot	58	465	約 12.5%
sequential	426	465	約 91.6%

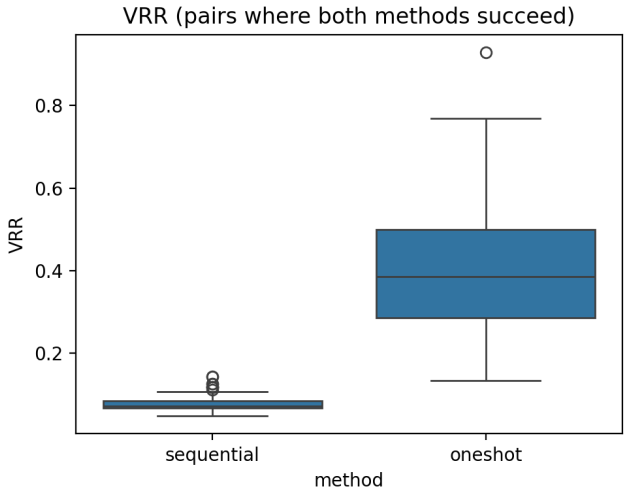
付録 2 制約遵守率(合計)

制約	手法	成功数	総数	成功率
easy	oneshot	54	195	約 27.7%
easy	sequential	193	195	約 99.0%
medium	oneshot	4	270	約 1.5%
medium	sequential	233	270	約 86.3%

付録 3 制約・手法別の制約遵守率

手法	平均	中央値
oneshot	0.69	0.61
sequential	6.73	2.63

付録 4 実行時間（秒）の集計



付録 5 両方式が成功したパターンにおける VRR の分布