

卒業論文 2024 年度(令和6年度)

入力コンテキストを切り替えることによる 日本語入力高速化手法の提案

慶應義塾大学 環境情報学部

石川湧馬

増井俊之研究会

2025年1月

入力コンテキストを切り替えることによる 日本語入力高速化手法の提案

論文要旨

現状の日本語入力では、同音異義語や煩雑な変換フローに起因するユーザー負担の増大という課題感がある。

そこで本研究は、入力コンテキストを切り替えることで変換候補を最適化し、入力効率を向上させるシステムを提案する。これにより、変換ミスの低減と入力手間の削減、定型文の高速入力を実現する。

キーワード

日本語入力, IME, ユーザーインターフェース

目次

第1章 はじめに	1
1.1 研究背景	1
1.2 研究目的	1
1.3 本稿の構成	1
第2章 既存システムとその課題点	2
2.1 従来の日本語入力フロー	2
2.2 入力負荷の問題	2
2.2.1 同音異義語の問題	2
2.2.2 定型文入力の問題	2
第3章 提案手法の概要	3
3.1 コンテキスト切り替えの概念	3
3.2 コンテキストの登録	3
3.3 入力方法	3
第4章 システムの提案	4
4.1 開発環境と技術要件	4
4.2 インターフェース概要	4
4.3 プロトタイプの機能構成	4
第5章 評価と考察	5
5.1 入力速度の評価	5
5.2 コンテキスト切り替えの議論	5
5.3 コンテキスト作成コストの議論	5
5.4 今後の課題	5
第6章 結論	6

6.1 成果と知見	6
6.2 今後の課題	6
6.3 結論	6
謝辞	8
参考文献	8

図目次

図2-1 MacOSの日本語入力フロー 「ありがとう」	2
図2-2 同音異義語の変換例 「後攻」	3
図2-3 定型文入力の例 「お世話になっております」 (都度変換)	4
図2-4 定型文入力の例 「お世話になっております」 (一括変換)	4

第1章 はじめに

現状の日本語入力では、同音異義語やかな漢字変換フローに起因するユーザー負荷の増大という課題感がある。そこで本研究は、入力コンテキストを動的に切り替えてからユーザーに入力を開始してもらうことで変換候補を最適化し、入力効率を向上させるシステムを提案する。

1.1 研究背景

従来の日本語入力システムは、多数の候補から目的の言葉を選択する必要があるため、ユーザーに負担をかける場合がある。特に使う語彙が限られた業務上のやりとりなどでは、明らかに入力しないであろう語が選択肢に含まれてきてしまったり、単語が長く系統的に絞り込みは簡単にできるにも関わらずユーザーは全て正しくタイピングを行わないと変換できなかったりする。このようなシステムはユーザーの入力負担が大きく、思考を妨げる要因ともなっている。この問題に対処するため、新たなアプローチとして、入力時のコンテキストをに事前に切り替えることで、変換候補を最適化する手法を提案する。

1.2 研究目的

入力コンテキストの切り替えによる日本語入力的高速化と誤変換の低減を主眼に置き、プロトタイプを開発し評価を行う。評価範囲としては、主に文章入力シーンを想定し、既存システムとの比較して評価を行う。

1.3 本稿の構成

第2章では既存システムの課題を整理し、第3章で提案手法を概説する。第4章で実装したプロトタイプと機能、第5章で評価と考察を示す。最後に第6章で本研究の総括を行う。

第2章 既存システムとその課題点

本章では、従来の日本語入力システムの問題点を整理し、提案手法の背景となる課題を明らかにする。

2.1 従来の日本語入力フロー

MacOSに標準搭載されている日本語入力システムを例に取り、従来の日本語入力フローを説明する。一般的な日本語IMEの操作手順 「ありがとう」という単語を入力する場合を例にとる。以下は左からa,r,iと入力した時のIMEの変換候補の図である。

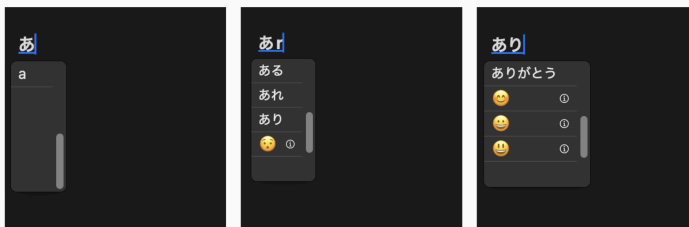


図2-1 MacOSの日本語入力フロー 「ありがとう」

「あり」が入力完了した時点で候補に「ありがとう」が表示された。スペースキーまたは下矢印キーを押すことで候補を確定し、変換が完了する。

2.2 入力負荷の問題

既存の日本語入力システムにおける課題点として、同音異義語と定型文入力の問題が挙げる

2.2.1 同音異義語の問題

「ありがとう」のような簡単な単語では問題はないが、複数の候補が表示される場合や、同音異義語が多い場合、ユーザーは候補を選択するために多くの操作を行わなければならない。

「後攻」という単語を入力する場合を考える。MacOSでは「こうこう」と入力すると、以下のような候補が表示される。



図2-2 同音異義語の変換例「後校」

この場合、ユーザーは候補を選択するために、スペースキーを押すか、下矢印キーを押して候補を選択する必要がある。今回であれば「こうこう」と入力してからスペースキーを7回押すことで「後校」を選択することができたが、このような操作はユーザーにとって負担が大きく、思考のスピードと日本語入力との間に大きく乖離が生じる。

これにより、入力効率が低下し、ユーザー負担が増大する。また、変換ミスが発生するリスクも高まる。

2.2.2 定型文入力の問題

また、定型文や挨拶文など、特定の文脈で頻繁に使用されるフレーズを入力する際にも、従来のIMEではその文章の読み方を入力する必要があるので、長い文章を入力したい場合には必然的に必要な工数が増加する。

「お世話になっております」というフレーズを入力する場合を考える。この場合、ユーザーは「おせわになっております」と入力する必要があるが、このような長いフレーズを入力する際には、ユーザーは入力に時間を要する。

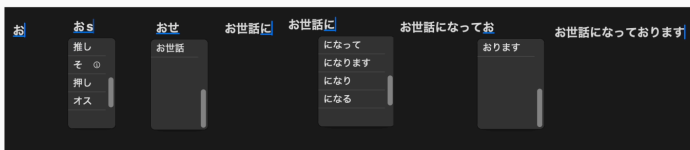


図2-3 定型文入力の例「お世話になっております」(都度変換)

「おせ」の時点で変換に「お世話」が表示されたので選択、その後に「に」を入力すると文脈を読んで「お世話」に接続する「に」から始まる候補として「になって」が表示されるので選択、その後に「お」を入力すると「お世話になって」に接続する「お」から始まる候補として「おります」が表示されるので選択し「お世話になっております」の入力が完了した。

文脈を読んで候補を表示するIMEの機能は非常に優れているが、このような長いフレーズを入力する際には、ユーザーは入力に時間を要する。

画像のような形態素ごとに変換を確定していく方法とは別で、「おせわになっております」と入力してからスペースキーを押すことで一気に変換する方法もあるが、この場合もユーザーは全ての読みを入力する必要があるため、入力効率が低下する。また、タイプミスリスクも高まるので入力効率の期待値は低い。

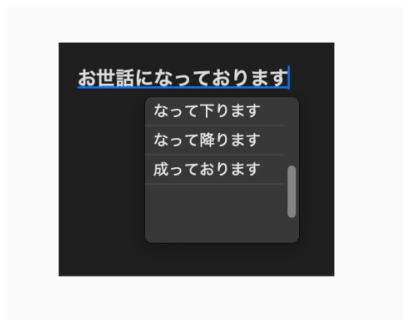


図2-4 定型文入力の例「お世話になっております」(一括変換)

第3章 提案手法の概要

第2章で整理した課題に対処するため、入力コンテキストを切り替えることで変換候補を最適化する手法を提案する。

3.1 コンテキスト切り替えの概念

まず、本稿におけるコンテキストとは何かを説明する。コンテキストとは、入力時における文脈や状況を指す。例えば、メールの文面を入力する際と、プログラムのコードを入力する際には、使用する語彙や文法が異なる。このような文脈を切り替えることで、IMEが適切な変換候補を表示することが可能となる。コンテキスト粒度は自由に設定可能なものであり、ユーザーが任意のコンテキストを登録し、切り替えることができる。「メール文面」と「プログラムコード」のような荒めのコンテキストから、「挨拶文」と「業務報告文」のような細かいコンテキスト、さらには「都道府県名」や「プロジェクト名」のような極めて細かいコンテキストまで設定可能な概念である。

3.2 コンテキストの登録

ユーザーは自分が使いたいコンテキストを事前に登録しておくことができる。登録しておいたコンテキストは、IMEの入力時に切り替えるための操作をすることで入力候補がそのコンテキストに合わせて表示されるようになる。

3.3 入力方法

変換ロジックが大きく異なるが、UIは従来のもとは大きく変わらない。ユーザーは現在のコンテキストの候補に入力することでそのコンテキストにおける候補一覧ロジックから出力された候補が表示される。ユーザーはその候補を選択することで変換を行うことができる。

第4章 システムの提案

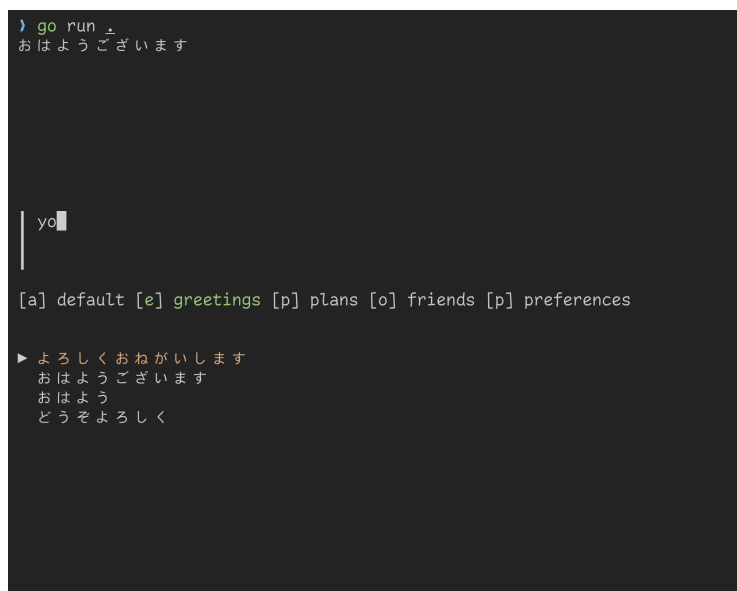
提案手法を検証するために開発したプロトタイプの開発環境や主要機能について概説する。

4.1 開発環境と技術要件

Go言語のTUIフレームワークであるbubbletea[1]とそのコンポーネントであるbubbles[2]を用いてターミナルで動作する今回提案するシステムのプロトタイプを開発した。Go言語はクロスプラットフォームで動作するため、Linux, MacOS, Windowsなどの環境で動作することが可能である。今回はMacOS Sequoia 15.1.1 で開発と動作確認を行った。

4.2 インターフェース概要

プロトタイプのインターフェースは以下のようになっている。



```
> go run .  
おはようございます  
  
yo  
  
[a] default [e] greetings [p] plans [o] friends [p] preferences  
  
▶ よろしくおねがいします  
  おはようございます  
  おはよう  
  どうぞよろしく
```

図4-1 システムインターフェース

1. 入力を確認した文字列を表示するエリア。
画像では「おはようございます」と入力した場合の画面を示している。
2. 現在入力中の文字列を表示するエリア。
画像では「yo」と入力中の場面を示している。
3. コンテキストの一覧と現在のコンテキストを表示するエリア。
画像では「[e] greetings」というコンテキストを選択している。
「[e]」というのはコンテキストのショートカットキーであり、
「greetings」というのは挨拶文コンテキストであることを示している。
4. 現在のコンテキストにおける変換候補を表示するエリア。
画像では「よろしくお願いします」「おはようございます」「おはよう」「どうぞよろしく」の4つの候補が表示されている。

4.3 プロトタイプの機能構成

ユーザーが複数のコンテキスト辞書を登録できる仕組みや、スニペット登録画面、コンテキストの自動切り替えロジックなどを搭載している。最小限の機能実装で効果を測定できるよう設計した。

第5章 評価と考察

提案手法の有効性を検証するために、小規模かつ予備的な評価を実施した。本章では、その概要と得られた知見を示し、今後の発展可能性について考察する。

5.1 入力速度の評価

本研究では、限られた参加者数と短いタスクを用いた予備的評価を行い、入力速度や変換精度の変化を観察した。また、操作性や学習コストなどの主観的評価もアンケート形式で収集し、提案手法の基本的な有効性や課題を洗い出すことを目的とした。

5.2 コンテキスト切り替えの議論

評価規模が限定的であったため、定量的な結果の精度は十分ではないが、入力速度が向上する傾向や変換精度の向上が見受けられた。主観的評価においても、コンテキスト切り替え機能が有効であるとの肯定的な意見が多かった。一方で、操作方法への慣れが必要という意見や、利用シーンの限定を指摘する声もあり、より大規模で多様な環境を想定した評価が今後の課題として挙げられる。

5.3 コンテキスト作成コストの議論

今回の評価はあくまで提案手法の概念実証を目的としたものであり、データ規模や参加者属性に限界があった。今後は多様な利用ケースを想定したタスク設計や、従来システムとの詳細な比較実験を実施し、定量的な入力効率やユーザー満足度をより厳密に検証する必要がある。また、学習コストの軽減やUI/UXの改善を行い、実運用に耐えるシステムに発展させることを目指す。

5.4 今後の課題

今回の評価はあくまで提案手法の概念実証を目的としたものであり、データ規模や参加者属性に限界があった。今後は多様な利用ケースを想定したタスク設計や、従来システムとの詳細な比較実験を実施し、定量的な入力効率やユーザー満足度をより厳密に検証する必要がある。また、学習コス

トの軽減やUI/UXの改善を行い、実運用に耐えるシステムに発展させることを目指す。

第6章 結論

本研究で提案したコンテキスト切り替えによる日本語入力高速化手法の成果と、残された課題について振り返る。

6.1 成果と知見

入力コンテキストの動的切り替えによって従来の日本語入力抱える多大な変換負荷を軽減できることが明らかになった。また、スニペット活用との併用により入力時間の大幅削減も期待できる点が確認された。

6.2 今後の課題

今後はユーザー行動の予測精度向上や、異なるプラットフォームへの適用可能性の調査が必要である。また、辞書学習の自動化やインターフェースのさらなる使いやすさの追求も残されている課題の一つとなる。



6.3 結論

本稿では、入力コンテキストを切り替えることによる新たな日本語入力手法を提案し、プロトタイプ評価からその有効性を示した。同音異義語変換ミスの削減や入力時間短縮に寄与することが明確となり、さらなる発展に向けた基礎を築いた。今後の拡張と実用化に向けて、ユーザー視点を取り入れた機能改善を継続して行う必要がある。

謝辞

本研究の実施にあたり、多大なるご支援とご協力をいただいた増井俊之教授、TAとして助言をいただいた尾崎正和氏に深く感謝いたします。

参考文献

1. A powerful little TUI framework 
<https://github.com/charmbracelet/bubbletea>
2. TUI components for Bubble Tea 
<https://github.com/charmbracelet/bubbles>