

修士論文 2014 年度（平成 26 年度）

ユーザのコンテキストを利用したテキスト入力 システムの研究

慶應義塾大学大学院 政策・メディア研究科

臼杵 壮也

インタラクションデザインプロジェクト

2015 年 1 月

修士論文 2014 年度（平成 26 年度）

ユーザのコンテキストを利用したテキスト入力システム の研究

論文要旨

モバイルデバイスの進化と普及と共に文字入力への機会は増加している。しかし現状スマートフォンを利用している人々の日本語入力 IME には大きな変化はみられない。そこで本研究では現状のモバイルデバイスに合った、より楽しく早い入力を可能にする日本語入力 IME 「Rive 日本語入力」を設計し、開発した。本システムにおける最大の特徴はコンテキストによる集合知を使った候補単語の推薦を行うことにある。コンテキストを使用することによって、ユーザは意識することなくその状況に適した入力候補単語を受け取り、選択することで入力を完了する。開発した推薦エンジンの設計や実装を述べると共に採用したユーザインタフェースを提示することで、既存の IME にはない有用性を述べる。また本システムにおける課題とそれに対するアプローチを述べることで、日本語入力における本研究の意義を論ずる。

キーワード

IME、日本語入力、コンテキスト、推薦

慶應義塾大学大学院 政策・メディア研究科

白杵 壮也

Abstract Of Master's Thesis Academic Year 2014

Development of a context-based text input system

Summary

Eigo ga dekinai node Roma-ji de soreppoi hunniki wo daseruto iina.

Murippoi desu ne.

Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here.

Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here.

Write down your abstract here. Write down your abstract here.

Keywords

Input Method Editor, Japanese Input, Context, Recommend

Graduate School of Media and Governance
Keio University

Usuki Masaya

目 次

| | | |
|-------|----------------------|----|
| 第 1 章 | 序論 | 1 |
| 1.1 | 研究の目的 | 2 |
| 1.2 | 用語定義 | 2 |
| 1.3 | 本論文の構成 | 3 |
| 第 2 章 | 背景 | 5 |
| 2.1 | モバイルデバイスへの文字入力のお会の増加 | 6 |
| 2.1.1 | モバイルデバイスの普及 | 6 |
| 2.1.2 | モバイルデバイスの用途 | 6 |
| 2.1.3 | デバイスの多様化 | 8 |
| 2.2 | 日本語入力と英語入力の差異 | 8 |
| 2.3 | 入力システムの変遷 | 8 |
| 2.4 | 日本語 IME への不満と期待 | 9 |
| 2.4.1 | 日本語 IME の不満 | 9 |
| 2.4.2 | IME への期待 | 10 |
| 第 3 章 | 設計 | 11 |
| 3.1 | Rive 日本語入力システム | 12 |
| 3.2 | Rive Client | 12 |
| 3.3 | Rive Server | 14 |
| 3.4 | Rive Analytics | 14 |
| 3.5 | Rive Batchprocessing | 14 |
| 3.6 | Rive Webservice | 14 |
| 3.7 | システム間通信 | 14 |
| 第 4 章 | 推薦システム | 15 |
| 4.1 | 概要 | 16 |
| 4.2 | 取得コンテキスト | 16 |
| 4.2.1 | 静的コンテキスト | 16 |
| 4.2.2 | 動的コンテキスト | 17 |
| 4.3 | Jubatus | 18 |
| 4.3.1 | 概要 | 18 |
| 4.3.2 | アルゴリズム | 18 |

| | | |
|--------------|-----------------------------------|-----------|
| 4.4 | 推薦例 | 19 |
| 第 5 章 | 実装 | 21 |
| 5.1 | Rive Client | 22 |
| 5.1.1 | システムフロー | 22 |
| 5.1.2 | コンテキスト取得 | 23 |
| 5.2 | Rive Server | 23 |
| 5.2.1 | システム構成 | 23 |
| 5.2.2 | Routing Server | 24 |
| 5.2.3 | beforeRules | 24 |
| 5.2.4 | Jubatus | 24 |
| 5.2.5 | afterRules | 24 |
| 5.3 | Rive Analytics | 24 |
| 5.3.1 | 指標値に用いる単語の説明 | 25 |
| 5.3.2 | 指標値 | 25 |
| 5.4 | Rive Batchprocessing | 26 |
| 5.5 | Rive Webservice | 26 |
| 第 6 章 | ユーザーインタフェース | 29 |
| 6.1 | キーボード | 30 |
| 6.2 | ダブル辞書インタフェース | 30 |
| 6.3 | 単語変換機能 | 30 |
| 6.4 | 候補ビューの横スクロール | 30 |
| 第 7 章 | 課題 | 33 |
| 7.1 | コールドスタート問題 | 34 |
| 7.2 | 省入力化と多様性のジレンマ | 34 |
| 7.3 | プライバシー問題 | 34 |
| 7.3.1 | 法律的問題 | 34 |
| 7.3.2 | 心理的問題 | 35 |
| 第 8 章 | 関連研究 | 37 |
| 8.1 | IME に関する研究・製品 | 38 |
| 8.1.1 | コンテキストアウェア IME の実現へ向けた動的辞書生成手法の提案 | 38 |
| 8.1.2 | Social IME | 38 |

| | | |
|---------------|-----------------------------|-----------|
| 8.1.3 | iWnn | 38 |
| 8.1.4 | Google 日本語入力 | 38 |
| 8.1.5 | 各種クライアント | 39 |
| 8.2 | コンテキストを使用した推薦システム | 39 |
| 8.2.1 | コンテキストの分類 | 39 |
| 8.2.2 | | 39 |
| 第 9 章 | 結論 | 41 |
| 9.1 | 研究から得られた成果 | 42 |
| 9.2 | 本システムの展望 | 42 |
| 9.3 | 研究への考察 | 42 |
| 第 10 章 | 本研究に関連する発表 | 43 |
| 10.1 | 展示会 | 43 |
| 10.2 | その他 | 43 |
| | 謝辞 | 44 |
| | 参考文献 | 46 |

図 目 次

| | | |
|-----|---|----|
| 2.1 | 情報通信端末世帯保有率の推移 (出典:[18]) | 6 |
| 2.2 | スマートフォンでどのような機能・アプリを使っているか (出典:[6]) | 7 |
| 2.3 | スマートウォッチの例:Android Wear を搭載した Moto360(出典:[1]) | 8 |
| 2.4 | Google glass(出典:[2]) | 8 |
| 2.5 | スマートフォンを利用している際に不便だと感じる点 (出典:[16]) | 9 |
| 2.6 | 折りたたみ式携帯電話例:(出典:[4]) | 10 |
| 2.7 | ストレート式携帯電話例 (出典:[4]) | 10 |
| 2.8 | スライド式携帯電話例:(出典:[4]) | 10 |
| 3.1 | システム全体構成図 | 12 |
| 3.2 | Rive Client ユーザ画面 | 13 |
| 5.1 | Rive Client フローイメージ | 22 |
| 5.2 | Rive Server 概要図 | 23 |
| 5.3 | Rive Analytics 画面 | 26 |
| 5.4 | Rive Webservice の体験画面 | 27 |
| 6.1 | qwerty キーボード | 30 |
| 6.2 | flick キーボード | 30 |
| 6.3 | 単語フリックイメージ図 | 31 |

第1章 序論

本章では研究の目的、使用する用語の定義と論文の構成について述べる。

1.1 研究の目的

本研究の目的は世の中の日本語を入力する人々がコンピュータあるいはモバイルデバイスへの入力を行う際に、ユーザの文字入力における楽しさを向上させると共に文字入力の速度を向上させることが目的である。その目的を達成するために Rive 日本語入力というシステムを実装・開発した。

1.2 用語定義

本論文において使用する用語を以下のように定義する。

デバイス

本論文に置いてデバイスとは単にハードウェアを意味している。

モバイルデバイス

本論文において、モバイルデバイスとは持ち運び可能であり、インターネットに接続可能な端末のことを意味している。

スマートフォン

本論文において、スマートフォンとは多機能なモバイルデバイスであり、パソコンとしての機能を果たせるものを意味している。具体的には Android,iOS などを搭載しているものである。

タブレット

本論文において、スマートフォンと同じ要件を満たしながら、画面のサイズがスマートフォンより大きい物を意味している。

IME

IME とは Input Method Editor の略であり文字入力補助ソフトと言われる。。コンピュータなどの情報機器において文字入力をする際に、それを手助けするシステムのことを IME と呼ぶ。

コンテキスト

コンテキストとは本論文において状況という意味で使っている。

集合知

集合知とは複数のデータを集める事によって新たな価値を出すという意味で使っている。

省入力

本論文において省入力とは入力の時間または手間を省くことを示している。

候補単語

本論文に置いて候補単語とは入力における、変換候補と推薦候補の二つを合わせた表現である。

クライアント

本論文においてクライアントとはユーザーが直接利用するアプリケーションあるいはそのデバイスを指す表現である。

1.3 本論文の構成

第2章では Rive 日本語入力の開発に至った背景について述べる。第3章では Rive 日本語入力の設計について述べる。第4章では推薦システムの使用について述べる。第5章では Rive 日本語入力の実装について述べる。第6章では Rive 日本語入力のユーザーインターフェースについて述べる。第7章では Rive 日本語入力の課題について述べる。第8章では関連研究を述べる。第9章では本研究の結論を述べる。最後に第10章では研究に関する発表について述べる。

第2章 背景

本章では本研究の背景となった、モバイルデバイスへの文字入力機会増加、入力のシステム歴史、文字入力への問題について述べる。

2.1 モバイルデバイスへの文字入力の際の増加

モバイルデバイスにおいて、文字入力をする機会が増加していて、IME の省入力化への需要が高まっているが、それにはいくつかの要因がある。

2.1.1 モバイルデバイスの普及

現在、日本においてスマートフォンやタブレットなどのモバイルデバイスが大幅かつ急激に普及した。総務省による「平成 25 年通信利用動向調査」[18] の「通信端末世帯保有率の推移」(図:2.1)によると、平成 22 年には 9.7 % しかなかったスマートフォンの普及率は、平

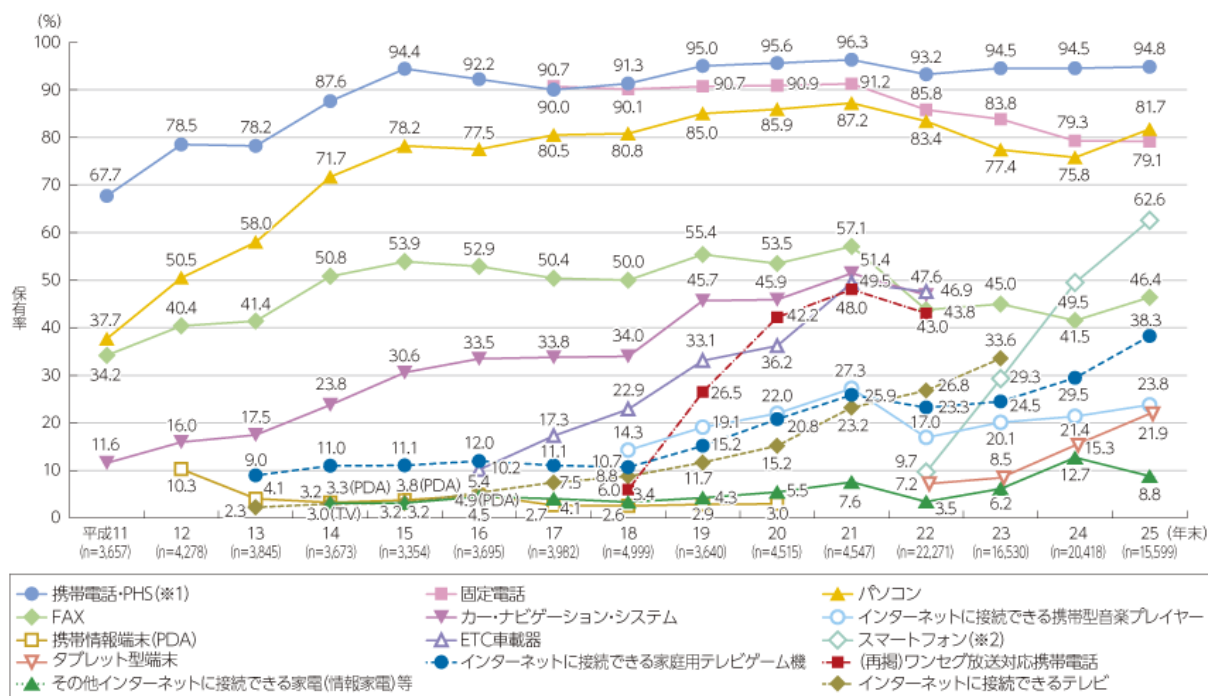


図 2.1: 情報通信端末世帯保有率の推移 (出典:[18])

成 25 年には 62.6 % と急激に成長している。平成 22 年から平成 25 年の 3 年間で 52.9 % の伸びを見せており、これはパソコンの保有率が最も伸びた平成 11 年から平成 14 年の伸び率を大きく上回る数字となっている。このデータからもスマートフォンの普及は未だかつてないほどの速度で進んでいることがわかる。

2.1.2 モバイルデバイスの用途

スマートフォンはパソコンの様に多くのことができるが、文字入力を行う可能性が高いアプリケーションは頻繁に使用されている。リサーチバンクによるアンケート調査 [6] (図:2.2) によると メールアプリを使う人が 78.1 %、SNS アプリを使う人が 44.8 %、コミュニケー

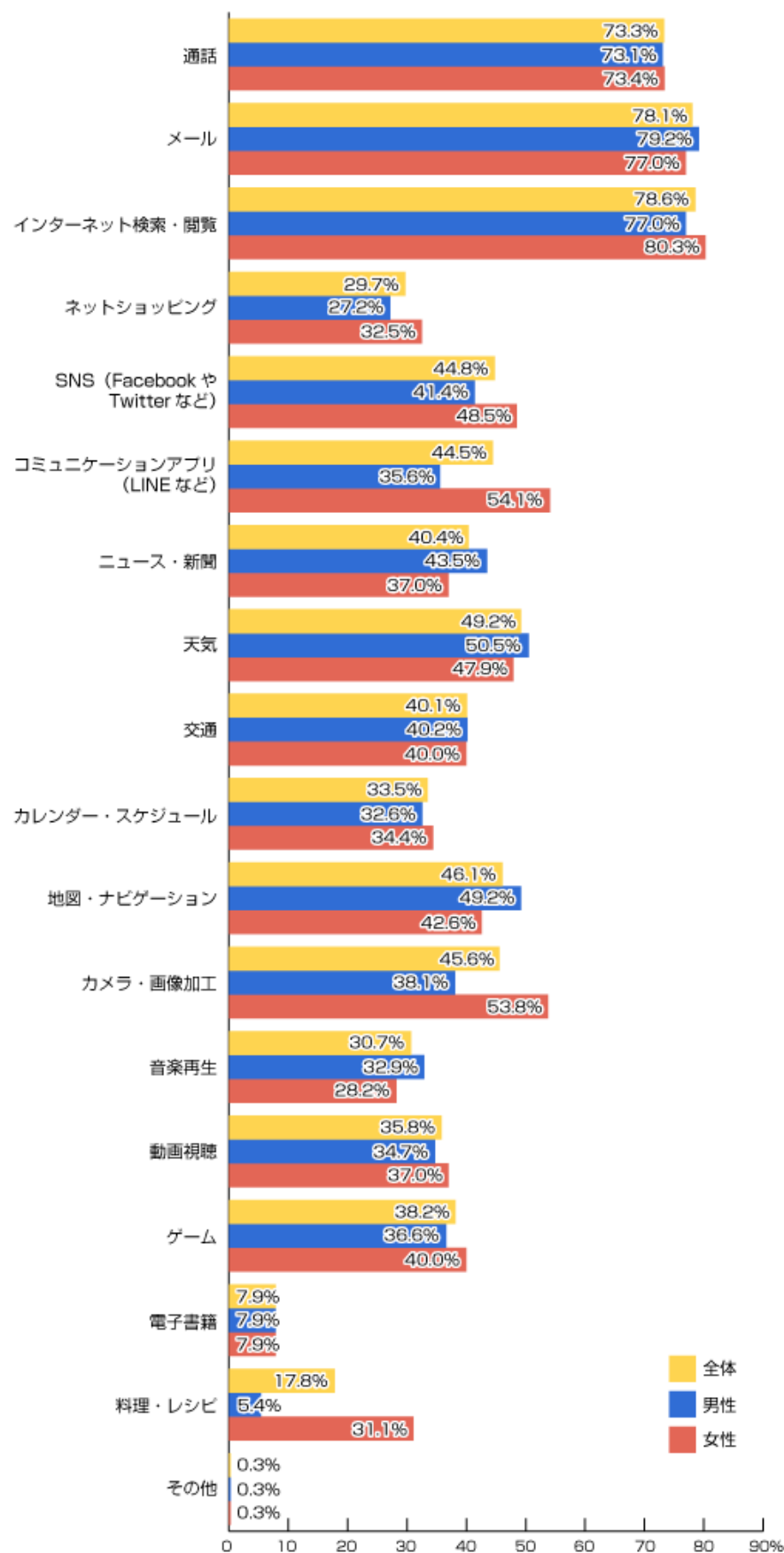


図 2.2: スマートフォンでどのような機能・アプリを使っているか (出典:[6])

ションアプリを使う人が 44.5 %となっている。メールアプリ、コミュニケーションアプリ、SNS アプリはそれぞれ情報の受信端末としてだけ使う場合には文字入力が必要はないが、情報を発信しようとする場合には文字入力が必要である。これら以外の他のアプリにおいても適時文字入力が必要である。つまりスマートフォンを使うユーザーは文字入力の機会がとて多いことがわかる。

2.1.3 デバイスの多様化

デバイスが多様化し、文字入力の機会が増えている。スマートウォッチ (図:2.3) や、ヘッドマウントディスプレイの一つである Google Glass(図:2.4) がその例である。スマートウォッ



図 2.3: スマートウォッチの例:Android Wear を搭載した Moto360(出典:[1])



図 2.4: Google glass(出典:[2])

チは既存の時計にモバイルデバイス OS を搭載したもので、従来時計であれば文字入力とは無縁であったが、これを使うことで、文字入力をこのデバイス上でも行う必要が出てくる。

2.2 日本語入力と英語入力の差異

日本語と英語では入力にかかるコストが異なる。qwerty キーボードで日本語を入力する場合、アルファベットを入力し、それをひらがなに日本語 IME が変換し、それを漢字やカタカナの混ざった文章に変換するというプロセスをとる。英語においては入力したアルファベットがそのまま文章となるため、IME の必要がなく、ほとんど使われていないのが現状である。

2.3 入力システムの変遷

現在どのような OS を使用していても必ず IME アプリケーションは搭載されている。これらのモバイルデバイスを使う上で日本語の入力は欠かすことができない操作である。デバ

イスの性能は携帯電話の頃から劇的に向上している。パソコンに遜色ないようなメモリや CPU を積んでいるものも多く市販されている。その間に様々なテキスト入力方式などが考案された [19] が搭載されている入力方式には携帯電話の時から大きな変化はない。日本語 IME における内部的なシステムとしては、連文節変換や予測入力 [7] の導入がおこなわれ、広く普及したがいまだ改善の余地は多く残されていると考える。

2.4 日本語 IME への不満と期待

日本語 IME には様々な問題点が存在する。

2.4.1 日本語 IME の不満

楽天リサーチによる「スマートフォンに関する調査」(出典:[16])によると、スマートフォンを使うユーザーが挙げる不満点の中で最も割合が大きいのが「文字入力がしにくい」ということである。(出典:[16]) 文字入力がしにくい理由としてはデバイスが携帯電話から変化し

| | | n | 文字 入力が しにくい から | 文字 が見に くい | ボタ ンが押 しにくい | 画面 が小さ いから | 回線 が遅い から | スマ ホの操 作に不 安があ るから | 音声 が聞き 取りに くい | 使い 方がわ からない | 電池 消費が 早い | 設定 がわか りにく い | 重い | 軽い | その他 | 特に 不便だ と感じ ることは ない |
|----|-----|------|-------------------------|-----------------|-------------------|------------------|-----------------|--------------------------------|------------------------|-------------------|-----------------|-----------------------|------|-----|-----|--------------------------------|
| 全体 | | 1000 | 53.3 | 32.0 | 32.3 | 48.1 | 18.4 | 6.0 | 2.7 | 4.1 | 50.8 | 8.7 | 7.6 | 0.1 | 1.8 | 9.0 |
| 性別 | 男性 | 500 | 57.0 | 34.0 | 37.8 | 45.8 | 18.2 | 3.6 | 2.8 | 4.2 | 53.2 | 8.2 | 7.0 | 0.2 | 0.8 | 8.4 |
| | 女性 | 500 | 49.6 | 30.0 | 26.8 | 50.4 | 18.6 | 8.4 | 2.6 | 4.0 | 48.4 | 9.2 | 8.2 | 0.0 | 2.8 | 9.6 |
| 年代 | 20代 | 200 | 46.5 | 19.0 | 26.0 | 31.5 | 19.0 | 4.0 | 2.0 | 1.0 | 48.5 | 4.5 | 7.0 | 0.5 | 2.0 | 16.0 |
| | 30代 | 200 | 56.0 | 21.5 | 38.0 | 46.5 | 22.5 | 4.5 | 2.5 | 2.0 | 50.5 | 4.0 | 10.5 | 0.0 | 1.0 | 9.5 |
| | 40代 | 200 | 58.5 | 33.0 | 37.5 | 45.0 | 21.0 | 2.5 | 4.0 | 3.5 | 55.5 | 7.5 | 8.5 | 0.0 | 3.0 | 8.0 |
| | 50代 | 200 | 54.0 | 44.0 | 31.5 | 58.5 | 17.5 | 9.5 | 2.0 | 6.5 | 47.0 | 12.0 | 5.0 | 0.0 | 2.5 | 6.0 |
| | 60代 | 200 | 51.5 | 42.5 | 28.5 | 59.0 | 12.0 | 9.5 | 3.0 | 7.5 | 52.5 | 15.5 | 7.0 | 0.0 | 0.5 | 5.5 |

図 2.5: スマートフォンを利用している際に不便だと感じる点 (出典:[16])

ているにもかかわらず、IME のインタフェースはほとんど変わっていないためであると考えた。携帯電話の時には折りたたみ式やストレート式、スライド式といくつか存在したが、入力方式はどれも同じであり、12 個あるキーをそれぞれ何度も押すことで、(三回タッチすると「あ-い-う」の用に变化する) 文字を入力していく方式であった。しかし現在のモバイルデバイスは多くがタッチスクリーン式に変化しており、これらキーパッド方式時代の名残



図 2.6: 折りたたみ式携帯電話例:(出典:[4])



図 2.7: ストレート式携帯電話例 (出典:[4])



図 2.8: スライド式携帯電話例:(出典:[4])

である入力方式はタッチスクリーンでの体験を重視したものに变化させていくべきである。
[11]

2.4.2 IME への期待

最終的には入力したいと思った文字をそのまま入力できるような IME を開発したいと考えている。その前段階としてコンテキストからユーザーの入力したい単語を推測し、提示することで入力のユーザーエクスペリエンスを大幅に向上させる IME を研究・開発した。また今回は日本語 IME として実装したが、実装されているコンテキスト推薦機能などは

第3章 設計

本章では Rive 日本語入力の全体の設計とその思想について述べる。

3.1 Rive 日本語入力システム

Rive 日本語入力システムは以下のものによって構成されている

- Rive Client
- Rive Server
- Rive Analytics
- Rive Batchprocessing
- Rive Webservice

これらのシステムがお互いに作用することで Rive 日本語入力システムが実現されている。

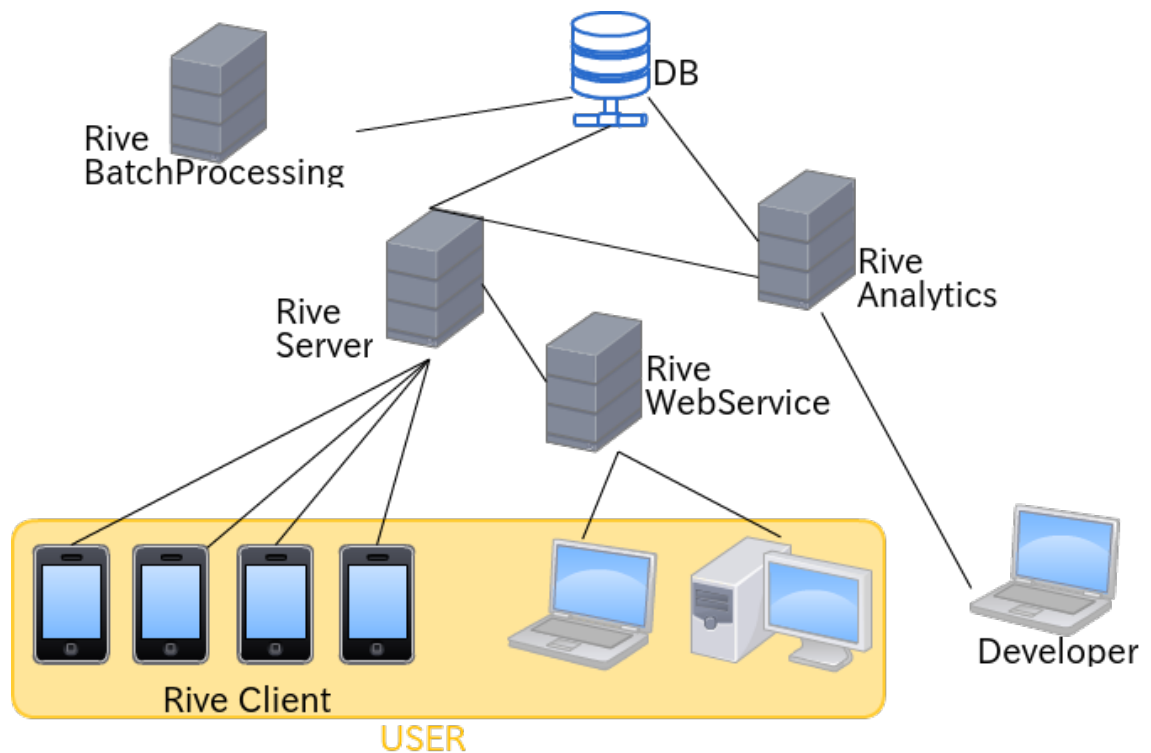


図 3.1: システム全体構成図

3.2 Rive Client

モバイルデバイスで利用可能な IME アプリケーション本体。ユーザーは基本的にこのアプリケーションのみ使用する。始めに Rive Server にコンテキストデータを送る。また帰ってきたデータを推薦単語としてユーザーに提示する。(Rive Client ユーザ画面:3.2)



図 3.2: Rive Client ユーザ画面

3.3 Rive Server

Rive 日本語入力における適切な候補単語を推測するサーバーの総称。Rive Client から受け取ったデータを解析し、適切な推薦を行う。また入力後のデータを Rive Analytics に送信する。

3.4 Rive Analytics

文字入力のデータを受け取り解析することで開発者の手助けをするシステム。バージョン管理システムである git¹ の commit² とデータを紐付けて有用性を確かめる為のシステム。ユーザは本システムへの直接の関わりは持たない。

3.5 Rive Batchprocessing

定期的に処理を行うシステムの総称。主に DB の更新、編集などを行う。

3.6 Rive Webservice

システムを試用するための WEB ページとして実装。また Rive 日本語入力の紹介も兼ねている。

3.7 システム間通信

Rive Client と Rive Server 間の通信は Websocket³、その他システム間は http によって実現した。Websocket を採用した理由はサーバー側からのプッシュ機構を導入するためと、http より速度が早いという理由である。[3]

¹<http://git-scm.com>

²<http://git-scm.com/docs/git-commit>

³<http://dev.w3.org/html5/websockets/>

第4章 推薦システム

本章では推薦システムについて述べる。

4.1 概要

ユーザーのコンテキストと入力を受け取り、過去の全ての人の入力情報と比べることによってユーザーの入力する単語を予測し推薦するシステム。クライアントを起動した瞬間から、一つの操作(キーボードにタッチすることや候補単語へのタッチすること)ごとに取得できる全てのコンテキストを入力と紐付けてサーバーへ送信して、推薦システムがそのコンテキストにあった推薦候補単語をクライアントへ送信するというプロセスを繰り返すことで成り立っている。

4.2 取得コンテキスト

入力が開始されてから終了まで変わらないコンテキスト何度も取得するとシステムに負荷がかかるため、変わるものとそうでないものに分類した。入力が開始してから変化しないコンテキストを静的コンテキスト。入力が開始してから変化する可能性があるコンテキストを動的コンテキストとした。コンテキストにはユーザーが設定するものと、Rive 日本語入力が自動的に取得するものがある。

4.2.1 静的コンテキスト

以下静的コンテキストとして利用しているものを述べる。

(1) アクティビティー

アクティビティーとは入力する対象のことである。メールクライアントであったり、WEBブラウザアプリなどである。アクティビティーをコンテキストに含めることにより、サービスごとに入力されやすい単語を推薦されやすくなる。

(2) 性別

性別をコンテキストに含めることにより、同じ性別の人が使っている単語を推薦されやすくなる。要素はユーザーが以下から選択する。

- 男性
- 女性
- その他

(3) 年齢

年齢をコンテキストに含めることにより、同じ年齢の人が使っている単語を推薦されやすくなる。

(4) キャラクター

キャラクターをコンテキストに含めることにより、簡易なユーザークラスタリングをし、同じクラスタの人が使っている単語を推薦されやすくなる。要素はユーザーが以下から選択する。

- ロタク
- JK
- 熱血
- 冷静
- おねえ
- 一般人

(5) Gmail アカウント

アカウントをコンテキストに含めることにより、ユーザーごとのパーソナライズされた単語を推薦されやすくなる。またこのコンテキストは極めてプライバシー性が高いため、デフォルトの設定では送信しないようになっている。

4.2.2 動的コンテキスト

以下利用している動的コンテキストについて述べる。

(1) 時間

時間をコンテキストに含めることにより、時間ごとに入力されやすい単語を推薦されやすくする。

(2) 日付

日付をコンテキストに含めることにより、日付ごとに入力されやすい単語を推薦されやすくする。

(3) 位置情報

位置情報をコンテキストに含めることにより、場所ごとに入力されやすい単語を推薦されやすくする。

(4) 加速度

加速度をコンテキストに含めることにより、加速度ごとに入力されやすい単語を推薦されやすくする。

(5) 現在の入力

現在未変換状態の文字が何であるかをコンテキストに含めることにより、どのような単語が求められているかを推測し推薦する。

(6) IME を起動してからの全ての入力

IME を起動してからの全ての入力をコンテキストに含めることにより、入力単語と同じ文章内で使われやすい単語を推薦されやすくなる。

(7) 入力の状態

IME を起動したタイミングなのか、文字を入力中なのか、変換または入力を確定したタイミングなのかという三つの状態のどの状態であるかを取得する。入力の状態をコンテキストに含めることにより、最初に入力されやすい単語や、終わりに入力されやすい単語などを推薦されやすくなる。

4.3 Jubatus

4.3.1 概要

Jubatus[8] とは株式会社 Preferred Infrastructure と NTT ソフトウェアイノベーションセンタが共同開発した、オンライン機械学習向け分散処理フレームワークである。本システムを使用することで推薦単語の候補計算をしている。Jubatus を Rive 日本語入力のシステムに組み込んだ理由は、高速かつスケーラビリティに優れていたためである。[15] Rive ではコンテキストを常に計算するため高速な処理が必要となり、採用した。

4.3.2 アルゴリズム

取得した単語一つ一つがベクトルとなるような多次元の特徴ベクトルを作っている。その n 次元上で取得したコンテキストから新たなベクトルを作り、近いものから順にスコア付し、候補単語として返すようになっている。機械学習アルゴリズムには Adapting Regularization of Weight Vectors[5] を使っている。

4.4 推薦例

本システムにおいて起こる推薦の事例について説明する。

- SFC にはじめてきた人が SFC ならではの話題をつぶやく場合
湘南台駅という過去の入力と SFC にいるという位置情報コンテキストを使うことで、「かもる」「諭吉像」「SFC」「ツインライナー」という推薦候補を得る。
- お盆休みに箱根に旅行にきた人が友人へ状況を説明する場合
お盆休みという日付情報と箱根という位置情報コンテキストを使うことで、「箱根」「強羅」「温泉」という推薦候補単語を得る。
- サッカーを見ている時に日本が点を決めたことを友人に報告する場合
LINE¹を開いているというコンテキストとインターネットにおいて最近盛り上がっている単語を使うことで、「ゴール」「日本」「先制」という推薦候補単語を得る。
- 朝起きて教授に遅刻の連絡をする場合
メールクライアントを開いているというコンテキストと起動時であるというコンテキストと自宅であるというコンテキストと昼という時間のコンテキストを使うことで、「申し訳ありません」「遅刻させていただきます」「よろしくお願いいたします」という推薦候補単語を得る。

¹<http://line.me/>

第5章 実装

本章では Rive 日本語入力を構成する各システムについて詳細な実装の解説を行う。

5.1 Rive Client

本システムは AndroidOS¹ 上で動くアプリケーションとして実装した。ユーザは本アプリケーションをインストールし、使用する IME に選択することで Rive 日本語入力を利用することができる。

5.1.1 システムフロー

このシステムが立ち上がるとまず始めに `onCreate()` が呼ばれ、初期化を行う。その後 `onStartInput()` メソッドが呼ばれコンテキストを取得する。このコンテキストはデバイスの情報を取得する。コンテキストを取得し次第、候補単語を取得する。この候補単語はサーバーと通信した上で取得する。これをユーザーが一つの操作を行うたびに繰り返す。ここで言う一つの動作とは、キーボード上の一つの文字を押すことや、候補の単語をタップすること、あるいは文字をデリートすることも含まれる。最終的にユーザーが入力を終了した場合には `onFinishInput()` が呼ばれ、今回のユーザが行った動作とコンテキストを紐付けサーバーに送信する。通信が終わり次第 `onDestory()` が呼ばれ本システムは終了する。(システムフローイメージ図:5.1)

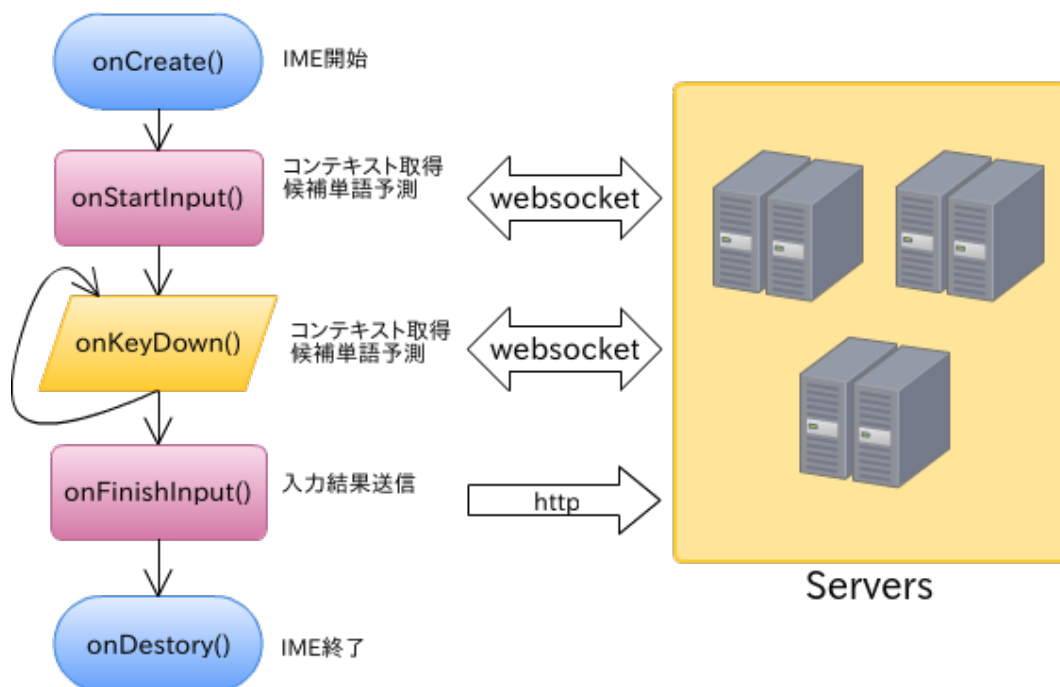


図 5.1: Rive Client フローイメージ

¹<http://www.google.com/intl/ja/mobile/android/>

5.1.2 コンテキスト取得

取得するコンテキストについては 4.2 項を参照。コンテキストはユーザーが入力できるものはユーザが入力する。その他デバイスで取得可能なものを全て取得し、推薦システムに使う。

5.2 Rive Server

VPS 上で動かしているサーバー群である。主に推薦に関する処理を担っている。

5.2.1 システム構成

サーバーの中身は大きく Routing Server, beforeRules, afterRules, Jubatus の 4 つを実装した。Routing Server は Node.js¹ によって実装した。それぞれ候補単語の計算手法が異なるため (図:5.2) ような分割になっている。始めにクライアントがコンテキストデータを Routing

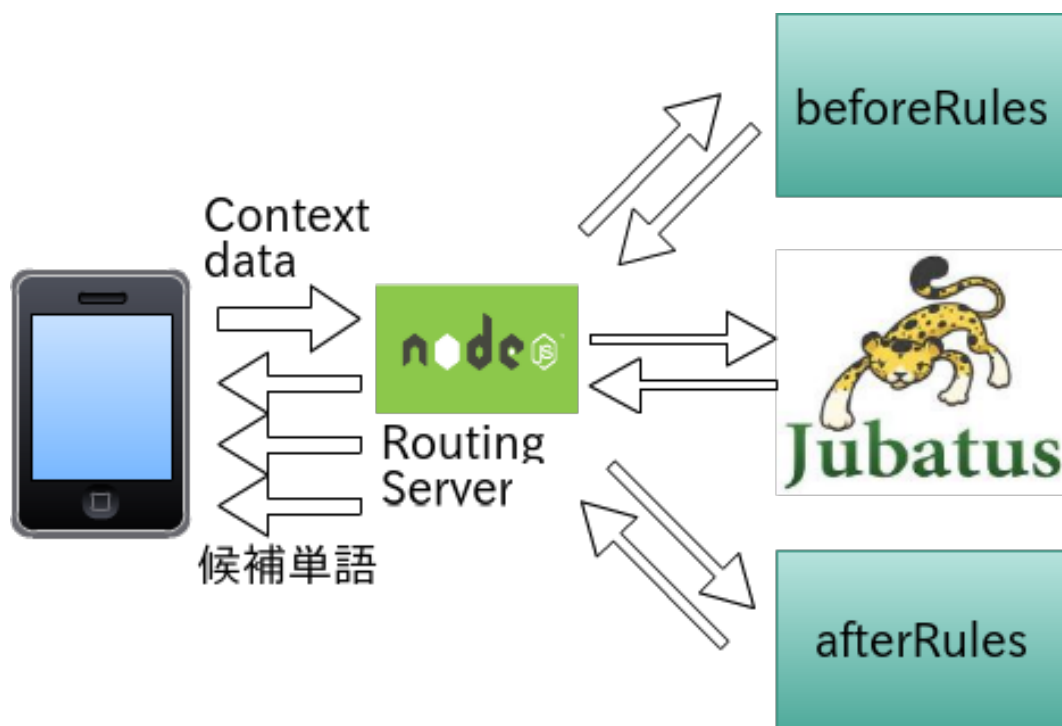


図 5.2: Rive Server 概要図

Server へ送信する。その後 Routing Server が beforeRules へコンテキストデータを送る。beforeRules は送られてきたデータを元に推薦候補単語を Routing Server に送る。Routing Server は受け取ったデータをクライアントに返すと共に、Jubatus へコンテキストデータ

¹<http://nodejs.jp/>

を再送する。Jubatus はそのデータを元に推薦候補単語を推測し Routing Server へ送る。Routing Server は受け取ったデータを更にクライアントへ送る。そしてそれと共にコンテキストデータを afterRules に再送する。そのデータを元に推薦候補単語を Routing Server へ送る。そしてそれをクライアントに返すことで完了する。一回のタッチごとにこれら全てのプロセスを行う。またこれらの一度の過程ごとにユニークな ID をつけてあり、途中で新しいプロセスが始まった場合前のプロセスは破棄されることで常に最新の推薦候補単語を受け取ることができる。

5.2.2 Routing Server

このサーバーはふたとおりの役割を持っている。一つはデバイスから受け取ったコンテキストデータを beforeRules, Jubatus, afterRules の 3 つの計算エンジンに振り分け送信する役割である。もうひとつは受け取ったデータを受け取り次第、デバイスに送信する役割である。

5.2.3 beforeRules

このエンジンはコンテキストデータが送られてきた際に、一定のルールに基づいているものをまとめている。このルールは開発者がよく使う単語を推測し実装した。例えば、Twitter² クライアントに入力を行っている（行おうとしている）場合に「なう」「@」の候補単語を返すようになっている。

5.2.4 Jubatus

このエンジンについては第 4.3 項参照。

5.2.5 afterRules

このエンジンは beforeRules より重要度が低く、また計算に時間がかかるものを推測して実装した。例えば、最寄り駅をクエリとしてその駅を通過している電車の線をその場で WEB から検索し推薦単語として返すことができる。

5.3 Rive Analytics

このシステムは Rive Client^{5.1} でのプロセスの終わりに送られてくるデータを受信し、解析するシステムである。開発者はこのシステムを使用することによって、新しい機能の有用性などを確かめることができる。

²<http://twitter.com/>

5.3.1 指標値に用いる単語の説明

click タッチ

touch 候補単語のタッチ

before beforeRules(5.2.3 項) からのタッチ

after afterRules(5.2.5 項) からのタッチ

jubatus Jubatus(4.3 項) からのタッチ

recommend beforeRules と afterRules と Jubatus の総和からのタッチ

score Jubatus において算出したスコア

5.3.2 指標値

- **CPI - Click Per Input**
この指標値は一度の入力に対して平均でどれくらいクリックしたかを示している値である。値を低く保つことで入力の省入力化を達成できる。
- **msPI - milliseconds Per Input**
この指標値は一度の入力に対して平均で何ミリ秒かかったかを示している値である。値を低く保つことで入力の少入力かを達成できる。
- **TPI - Touch Candidate Per Input**
この指標値は一度の入力に対して平均で何度候補単語をタッチしたかを示している値である。推薦エンジンの有用性を示す値となる。
- **TPC - Touch Candidate per Click**
この指標値はクリック回数のうち候補単語がタッチされた割合を示す値である。割合を把握することで、キーボードと候補のどちらを改善すべきかを示す値である。
- **RPC - Recommend Per Touch**
この指標値は候補単語をタッチされた回数のうちどれだけの割合で推薦候補をタッチされたかを示す値である。この値を高めることで推薦エンジンの有用性を示す。
- **Score - Score average by Jubatus**
この指標値は Jubatus による推薦単語がタッチされた時に、いくつかのスコアのものがタッチされたかを示す値である。beforeRules や afterRules の閾値の参考にする値である。
- **BPR - BeforeRules Per Recommend**
この指標値は推薦単語がタッチされたうちの beforeRules のタッチされた割合を示す値である。beforeRules の有用性を示す値となる。
- **APR - AfterRules Per Recommend**
この指標値は推薦単語がタッチされたうちの afterRules のタッチされた割合を示す値である。afterRules の有用性を示す値となる。

- JPR - Jubatus Per Recommend
この指標値は推薦単語がタッチされたうちの Jubatus からきた推薦単語のタッチされた割合を示す値である。Jubatus からくる推薦単語の有用性を示している。
- JPT - Jubatus Per Touch
この指標値は候補単語をタッチした回数のなかで Jubatus からくる推薦単語の割合を示す値である。jubatus からくる推薦単語の有用性を示している。

これらを git の commit とひもづけることによってバージョンごとにデータを管理している。
(図:5.3)

| commit | コメント | コミット者 | ビルド日時 | 集計数 | CPI | msPI | TPI | TPC | RPC | Score |
|--------|--------------------|----------|------------|-----|--------------------|--------------------|--------------------|--------------------|-----|---------------------|
| 5282c2 | (amend): | nikezono | 2014.06.14 | 336 | 11.711309523809524 | 48.98779605019539 | 5.096315120711563 | 5.096315120711563 | 0 | NaN |
| dd90a2 | 入力終了時postにstatic追加 | nikezono | 2014.01.28 | 198 | 6.914141414141414 | 145.7448141018616 | NaN | NaN | 0 | NaN |
| 96d3b3 | (merge): | masaya | 2014.03.10 | 114 | 19.19298245614035 | 92.87883889039152 | NaN | NaN | 0 | NaN |
| d3c8fa | MANIFEST | nikezono | 2014.06.11 | 95 | 9.705263157894738 | 1882.5850812972305 | 1.7245119305856833 | 1.7245119305856833 | 0 | NaN |
| 12f46f | #166のdiffTimeを追加した | masaya | 2014.02.10 | 81 | 12.864197530864198 | 363.4756640109054 | NaN | NaN | 0 | NaN |
| 6e74d9 | #304 | nikezono | 2014.05.20 | 75 | 22.186666666666667 | 777.8671413820132 | 0.5835336538461539 | 0.5835336538461539 | 0 | 0.02099537471760129 |
| b6f119 | Travis対策のp | nikezono | 2014.02.24 | 65 | 2.6615384615384614 | 163.49846898788837 | NaN | NaN | 0 | NaN |
| f2d760 | MANIFEST | nikezono | 2014.05.22 | 54 | 7.981481481481482 | 140.2463052494643 | 2.62877030162413 | 2.62877030162413 | 0 | NaN |

図 5.3: Rive Analytics 画面

5.4 Rive Batchprocessing

このシステムは定期的に処理を行うものを管理し実行している。インターネット上のデータをクロールし、DB を定期的にアップデートするプログラムなどが構成している。

5.5 Rive Webservice

このシステムはインターネット上で閲覧可能なホームページとして実装した。Rive Server と Rive Webservice は通信を行い、仮想のコンテキストを作り出すことで推薦の状態を確認することができる。(体験画面イメージ:5.4)

Rive日本語入力

デモ動画機能紹介体験ダウンロード

Playground

Riveの入力システムを体験しよう

年齢

25

タイミング

LAUNCH

性別

男性

アプリ名

com.twitter.android

過去の入力

東京

サッカー

ボール

読みがな

あそび

時間

10

性格

@ before Twitter 0.1

#桜trick after hotword 0.1

#sakuratruck after hotword 0.09000000000000001

#ojikan after hotword 0.08

#sol after hotword 0.07

桜Trick after hotword 0.060000000000000005

マック after hotword 0.05

図 5.4: Rive Webservice の体験画面

第6章 ユーザーインタフェース

文字入力を行うほとんどのユーザーはIMEを利用する際にアプリケーションごとにIMEを変えたりせず、あらゆるアプリケーションに対して同じIMEを利用する。そこで基本的な入力を行うことができ、かつ推薦機構の導入を実現しなければならない。そこでIME本来の機能を損なわないように Rive Client に独自のインターフェースを実装することで Rive 日本語入力システムを実現した。またこれらのユーザインタフェースは日本語独自の設計ではないため容易に外国語の IME への実装が可能である。

6.1 キーボード

モバイルデバイス使用者が使う入力のうち最も使用者が多い二つのキーボードを使用可能にした。キーボードは qwerty キーボードとフリック入力キーボードの二つを実装した。ユーザーは好みに合わせて二つのキーボードを使用することができる。(実装したキーボード図:6.1, 図:6.2)



図 6.1: qwerty キーボード



図 6.2: flick キーボード

6.2 ダブル辞書インタフェース

IME としての通常の文字変換もないと普段の使用に耐えることができないため、二つの変換システムを実装している。上段はコンテキストによる推薦システム??から推薦される候補単語で、下段は通常の IME に搭載されている辞書である。

6.3 単語変換機能

この機能は候補単語をフリック操作することで、フリックした候補単語を元に、新たな候補単語群を推薦する。現在上方向へのフリック操作では英語変換をし、下方向へのフリック操作では類語変換を行う。この機能にがあることによって推薦単語が無意味になる可能性を減らすと共に、入力の多様性を確保する。この単語変換機能は上下段両方に使用可能な機能である。この機能は連続で 사용할 ことが できるため、類語から類語へと連続で新しい候補を辿ることや、類語変換したものを英語へ変換するといったことが可能である。(単語フリックイメージ図:6.3)

6.4 候補ビューの横スクロール

上下段どちらかの候補ビューを横にスクロール際に、もう一方も連動して動かせるように実装した。候補ビューをスクロールさせる際にはどちらの段にも表示されていない単語を入



図 6.3: 単語フリックイメージ図

力したい場合なので、共にスクロールすることで入力したい単語の検索を助ける。スクロールする方向はもう一方の段のスクロールに対し、順方向と逆方向の二通りから選択可能である。

第7章 課題

本章では Rive 日本語入力における課題とその解決策について述べる。

7.1 コールドスタート問題

本システムは集合知を使ってレコメンドを行っているため、データが十分に集まるまで適切な推薦を行うことができないという問題である。本来こういった集合知を使ったシステムにおいては、別のところから持ってきたデータ等をシステム用に多少加工し、始めのデータとして使うことでこの類の問題を解決する。しかし Rive 日本語入力については一定のコンテキストが付加された入力データが必要であるがインターネット上には存在しないデータがある。そこで第 5.2 項において述べた、beforeRules と afterRules の実装を行った。これを実装したことによりデータが少ない状況でも一定の推薦候補単語を得ることができるようになった。

7.2 省入力化と多様性のジレンマ

Rive 日本語入力を使用し、候補単語のクリックだけで入力が完了するようになると本システムの最大の目的は達成される。しかし本システムを使うあらゆる人が同じ言葉あるいは単語を使うことになって日本語の多様性が失われてしまうのではないかという懸念が省入力化と多様性のジレンマである。この懸念はコンテキストへの勘違いから起こる問題であると考えられる。物理的なものだけだとこの問題を解決することはできないが、精神的なコンテキストを導入することによって解決すると考えている。精神的なコンテキストまで集めることで、精神的状況さえきちんと把握することができれば多様性を持った十分な推薦ができると考えている。そのため Rive 日本語入力においては精神的コンテキストをできる限り導入したいと考えている。現状でもキャラクターや年齢といった精神的コンテキストは使用しているが、いずれは物理的センシングを使って精神的コンテキストを把握することで、より推薦の精度を高め、入力の多様化も担保したいと考えている。また単語変換機能(第 6.3 項)における類語変換を行うことにより、ユーザーの入力の多様化を推進している。

7.3 プライバシー問題

Rive 日本語入力においては、入力のデータを全てコンテキストと共にサーバーに送ることとで、推薦候補を推測しているが、これはプライバシー的に良いのかという問題である。

7.3.1 法律的問題

法律的にこの問題に関連しているのが「個人情報の保護に関する法律」、通称「個人情報保護法」である。個人情報保護法第 2 条 1 項によると

個人情報とは生存する個人の情報であって、特定の個人を識別できる情報(氏名、生年月日等)を指す。これには、他の情報と容易に照合することができるこ

とによって特定の個人を識別することができる情報(学生名簿等と照合することで個人を特定できるような学籍番号等) も含まれる。

とされていて、Rive 日本語入力においては問題がないと考えられる。しかし入力全てをサーバーに送っているため、「慶應義塾大学所属の臼杵壮也」という文章などを打った場合には個人情報になってしまうという問題がある。現状、サーバー側で個人情報にあたると思われるデータについては学習しないようにすることで解決している。

7.3.2 心理的問題

入力の全てを送るため、ユーザーの心理的ハードルはとても高い。現状ではユーザーに入力データを送信しないという選択肢を与えることで対応している。

第8章 関連研究

本章では Rive 日本語入力に関連した研究について述べる。

8.1 IME に関する研究・製品

Rive 日本語入力と既存の研究や製品を比較することで、Rive 日本語入力の特色を述べる。

8.1.1 コンテキストウェア IME の実現へ向けた動的辞書生成手法の提案

荒川らが研究・提案した動的辞書生成手法である。[17] コンテキストを利用し、様々な人のデータをひもづけることによって一つの辞書を作るという点に置いて Rive 日本語入力と類似している。差異はコンテキストのリアルタイム性にある。Rive 日本語入力においてコンテキストは動的コンテキストと静的コンテキスト?? に分かれているが、この動的辞書生成においては静的コンテキストとして私が定義したものしか考えられていない。またこのシステムではコンテキストについて位置情報についてのみ考察しているが Rive 日本語入力では、複数のコンテキストを総合的に解析する手法について設計し開発した。

8.1.2 Social IME

奥野が開発したユーザ間で辞書を共有する IME である。[14] Rive 日本語入力において複数のユーザが一つの辞書を利用するという集合知を用いた辞書の発想は類似している。この IME において使われているシステム [13] では以前の入力などと言ったコンテキストが考慮されおらず、Rive 日本語入力ではそれらをコンテキストに導入することで、文章全体における共起表現などの推薦を可能にした。

8.1.3 iWnn

omron 社が開発した日本語 IME であり、[9] 現在 AndroidOS において多くの機種で標準搭載しているものである。携帯電話向けに実装されていて、時間情報やメールの送信相手の情報からコンテキストを推測し、適切な推薦を行う点に置いて、Rive 日本語入力と類似している。コンテキストについては開発者の決めたものから推測するため、未知のコンテキストなど柔軟なコンテキスト推測ができないところが、Rive 日本語入力システムと異なる。

8.1.4 Google 日本語入力

Google 日本語入力は Google 社によって開発された IME である。Google の検索システムと連動しており最新かつ膨大な情報を使った辞書が利用可能である。検索システムを使っているため、集合知を使いリアルタイムな情報を得たいという点に置いて Rive 日本語入力と類似している。その上で実現するためのアプローチが異なっている。Google 日本語入力ではコンテキストとして利用されるものが過去の入力だけになっている。

8.1.5 各種クライアント

Rive 日本語入力システムにおいてコンテキスト推測で、アクティビティー (第 4.2.1) ごとに特化型クライアントが存在する場合、それらはIME機能を備えていることが多い。Twitter クライアントであれば、リプライを送る際に相手のアカウント名が自動で入力されるようになる等の入力補助がある。

8.2 コンテキストを使用した推薦システム

コンテキストを利用したシステムに関する研究は始め、位置情報や時間情報利用したシステムが多かった。その後、そこにユーザー情報を加えたシステムの研究が行われた。更にその後には、他のコンテキストを加えたシステムの研究が行われた。[12]

8.2.1 コンテキストの分類

Rive 日本語入力においては 4 において述べたように実装上、動的コンテキストと静的コンテキストの 2 種類に分類したが、Strang らの研究 [10] によると以下に分類される。

- Key-Value Models
- Markup Schema Models
- Graphical Models
- Object Oriented Models
- Logic Based Models
- Ontology Based Models

しかしこの分類は結果を見ずえた上での分類であり、コンテキストがどのような推薦結果を提示するのか予測が難しい Rive 日本語入力におけるコンテキストの分類には適さなかった。

8.2.2

第9章 結論

本章では研究において得られた結論について述べる。

9.1 研究から得られた成果

本研究における成果を以下にまとめる。

- 日本語 IME における様々な改善点を明らかにした。
- 他の IME や外国語にも適応可能な新しいユーザーインターフェースを提唱した。
- コンテキストを使用するシステムの有用性を明らかにした。
- Rive 日本語入力というシステムを設計し開発した。

9.2 本システムの展望

Rive 日本語入力において情報が集まることにより、指数関数的に使い勝手が向上すると考えている。そのために必要なことはアクティブユーザの確保とシステムの安定した運営である。まずアクティブユーザの確保を行うために、よりユーザが使用した時の楽しさを向上させるシステムの導入を順次行っている。また安定した運営のために、サーバクライアントモデルからより少ない通信で済むようなモデルへの変更を考えている。またデバイスが新しいコンテキストをとることが可能になると、本システムも自動的に性能が向上するように設計したため、システムの有用性は向上し続けると考えている。推薦システムにおいてはコンテキストごとにどれほど推薦に寄与しているかを測ることで、より精度の高い推薦ができるようになると考えている。

9.3 研究への考察

Rive 日本語入力の考案から始まり、設計、実装した上での考察である。最新の技術を利用することで、既存の技術や設計では成し得なかった、システムを実装することができた。特に推薦候補を計算し、提示するために十分な速度を出すことができるのかは疑問であったが充分使用に耐えうる速度での実装をすることができた。また推薦候補単語や変換システムを使うことで、いままでにない新しい入力へのエクスペリエンスをユーザに提示することができた。ユーザインタフェースにおいては日本語にこだわらない、多言語への適応も容易なインタフェースを提示した。日本語 IME にはまだ多くの改善点を残し、それに対する一つの有用なアプローチとして研究を行うことができたと考えている。

第10章 本研究に関連する発表

10.1 展示会

- SFC OpenResearchForum2013 (2013,11,22 23 東京ミッドタウンホール 主催：慶應義塾大学 SFC 研究所)
- SFC OpenResearchForum2014 (2014,11,21 22 東京ミッドタウンホール 主催：慶應義塾大学 SFC 研究所)

10.2 その他

- 独立行政法人情報処理推進機構 2013 年度未踏 IT 人材発掘・育成事業テーマ名: コンテキストに応じた変換候補を提示するシステムの開発
- 情報処理 情報処理学会 2014 年 12 月 p1324 未踏の第 20 期スーパークリエイター達 竹内郁雄

謝辞

本研究に関して終始ご指導ご鞭撻を頂きました、慶應義塾大学環境情報学部の増井俊之教授には深く感謝いたします。また本研究の副査としてご意見、ご助言をいただきました清木康教授、小川克彦教授に深く感謝いたします。

また本研究において共に研究開発を行い苦楽を共にした、研究室の同輩である中園翔さんには深く感謝しております。

研究を進めるにあたり、様々な助言を頂いた KDDI 株式会社サービス企画本部クラウドサービス企画開発部長の藤井彰人に深く感謝いたします。

2年間の大学院生活を行う上で、共に励まし合い高め合ってきた増井俊之研究室における同輩の橋本翔さん、馬場匠見さん、田中優さん、郡山隼人さん、桜井雄介さん、並びに学部生の方々には深く感謝いたします。

私の大学院における研究生生活の中で、大きな成長の機会を与えてくださった未踏プロジェクト並びにその関係者の方々に深く感謝いたします。

最後に、私の大学院における生活を支えてくれた両親、祖父母、妹達に感謝します。本当にありがとうございました。

参考文献

- [1] Android Wear - Wikipedia. http://ja.wikipedia.org/wiki/Android_Wear/.
- [2] Google Glass - Wikipedia. http://ja.wikipedia.org/wiki/Google_Glass/.
- [3] Rest vs websocket comparison and benchmarks. <http://blog.arungupta.me/2014/02/rest-vs-websocket-comparison-benchmarks/>.
- [4] 携帯電話- Wikipedia. <http://ja.wikipedia.org/wiki/携帯電話>.
- [5] Alex Kulesza Koby Crammer and Mark Dredze. Adapting regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, 2009.
- [6] Lifemedia. スマートフォンに関する調査. http://research.lifemedia.jp/2014/04/140409_smartphone.
- [7] Toshiyuki Masui. An efficient text input method for pen-based computers. *CHI*, 1998.
- [8] PFI & NTT. Jubatus オンライン機械学習向け分散処理フレームワーク. <http://jubat.us/ja/>, 2011-2014.
- [9] omron. iWnn. http://www.omronsoft.co.jp/product_text/iwnn/.
- [10] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. *Ubicomp*, 2004.
- [11] Jenifer Tidwell. デザイニング・インターフェース 第2版. O'REILLY, 2011.
- [12] 奥健太. ユーザーコンテキストを考慮した情報推薦方式に関する研究. PhD thesis, 奈良先端科学技術大学院大学, 2009.
- [13] 陽奥野, 将文萩原. インターネットを用いた日本語入力システム (日本語処理・文法). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2009, No. 36, pp. 1–6, mar 2009.
- [14] 奥野陽. Social IME. <http://www.social-ime.com>.
- [15] 大輔岡野原. 大規模データ分析基盤 jubatus によるリアルタイム機械学習. 人工知能学会誌 = Journal of Japanese Society for Artificial Intelligence, Vol. 28, No. 1, pp. 98–103, jan 2013.
- [16] 楽天リサーチ株式会社. スマートフォン利用に関する調査. Technical report, 楽天リサーチ株式会社, 2014.

- [17] 荒川豊, 末松真司, 田頭茂明, 福田晃. コンテキストウェア ime の実現へ向けた動的辞書生成手法の提案. 情報処理学会論文誌, Vol. 52, No. 3, pp. 1033–1044, 3 2011.
- [18] 総務省. 平成 25 年通信利用動向調査. Technical report, 総務省, 2012.
- [19] 俊之増井. 携帯端末のテキスト入力方法. ヒューマンインタフェース学会誌 = Journal of Human Interface Society : human interface, Vol. 4, No. 3, pp. 11–24, aug 2002.