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る人々の日本語入力 IME には大きな変化はみられない。本研究では現状のモバイルデバイスに合った、より面白く早い入力を可能にする日本語入力 IME 「Rive 日本語入力」を設計し実装した。本システムにおける最大の特徴はコンテキストを用いた候補単語の推薦を行うことにある。コンテキストと入力を関連付け、過去の入力データと照らし合わせることで推薦候補単語を計算する。コンテキストを用いた推薦候補単語を利用することで、ユーザー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、それを選択することで入力を行う。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

臼杵 壮也

Abstract Of Master's Thesis Academic Year 2014

Development of a context-based text input system

Summary

We are increasing at an opportunity to input character with evolution and the spread of mobile devices. But smartphone user's inputting hardly changes with Japanese IME. In this research, I designed and implemented Japanese IME: "RiveJapaneseInput" which enables users to quick input and interesting input. The most feature in this system is recommend words from user's context. User can receive recommend candidate words suitable for the situation unconsciously. And he only chooses these words. I propose some interfaces in this system which can apply other systems. In this paper, I implemented Japanese IME and discuss advantage of this system.

Keywords

Input Method Editor, Japanese Input, Context, Recommend

Graduate School of Media and Governance
Keio University

Usuki Masaya

目 次

図 目 次

第1章　序論

本章では研究の目的と論文の構成について述べる。

1.1 研究の動機

昨今、インターネット環境の充実化と共に一般の人々は日本語の入力をいつでもどこでも行うようになった。いままでならば文字入力を滅多に行わなかった人々なども facebook[1] や twitter[2] といった SNS[3] や LINE[4] 等のコミュニケーションアプリの普及と共に入力を頻繁に行っている。しかし日本語 IME はできることがコンピュータにおける IME と変わらずモバイルデバイスの性能を最大限活かしているとは言いづらい。そこでモバイルデバイスの性能を最大限活かした、今までとは違った入力体験をしたいと考え、新しい日本語入力 IME の設計・開発を行った。

1.2 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面白さを向上させると共に文字入力の速度を向上させるシステムの実装と提案をすることである。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。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示していて、何時でもどこでも入力を行うということが前提となっている。

1.3 本論文の構成

本論文は全 9 章で構成される。
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 章では推薦システムの仕様について述べる。
第 4 章では Rive 日本語入力の設計と実装について述べる。
第 5 章では Rive 日本語入力のユーザインタフェースについて述べる。
第 6 章では Rive 日本語入力の課題について述べる。
第 7 章では関連研究を述べる。
第 8 章では本研究の結論を述べる。
最後に第 9 章では研究に関する発表について述べる。

[1] http://www.facebook.com/
[2] http://twitter.com/
[3] Social Networking Service の略称
[4] http://line.me/

第2章　背景

本章では本研究の背景となった、日本語 IME への不満と期待、モバイルデバイスへの文字入力機会増加、日本語入力と英語入力の差異について述べる。

2.1 日本語 IME への不満と期待

日本語 IME には様々な不満と期待が存在する。

2.1.1 日本語 IME への不満

楽天リサーチにの「スマートフォンに関する調査」(出典:[19]) 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 (図:2.1) 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。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ている

		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 2.1: 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:[19])

にもかかわらず、IME の入力方式インタフェースがほとんど変わっていないためであると考えられる。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、入力方式はどれもキーパッド式であり、12 個あるキー (あかさたなはまやらわ#*) をそれぞれ何度も押すことで文字を入力していく方式であった。例えば、三回タッチすると「あ→い→う」の用になり入力をする方式である。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変化させていくべきである [13] と考えている。

2.1.2 IME への期待

本来、入力とは入力する内容に意味があり、入力することそのものには意味がない。そこで入力に関するコストは少なければ少ないほど優れた IME と言える。そこで最終的には入

図 2.2: 折りたたみ式携帯電話例:(出典:[4])

図 2.3: ストレート式携帯電話例 (出典:[4])

図 2.4: スライド式携帯電話例:(出典:[4])

力したいと思った文字を何もせずに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。しかしそこまでにはまだまだ現状とは大きな隔たりがある。そこで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に関するコストを大幅に削減する IME を開発した。本論文において入力におけるコストの削減を省入力化という言葉を用いて言い換えている。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外国語にも応用可能なものとして開発を行った。

2.2 モバイルデバイスへの文字入力の機会の増加

今日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る。そのため IME の省入力化への需要が高まっている。それにはいくつかの要因があり、そのいくつかの要因についてここでは論ずる。

2.2.1 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」[22] の「通信端末世帯保有率の推移」(図:2.5) 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いまだかつてないほどの速度で進行していることがわかる。本論文においては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すことができるものを意味していて、具

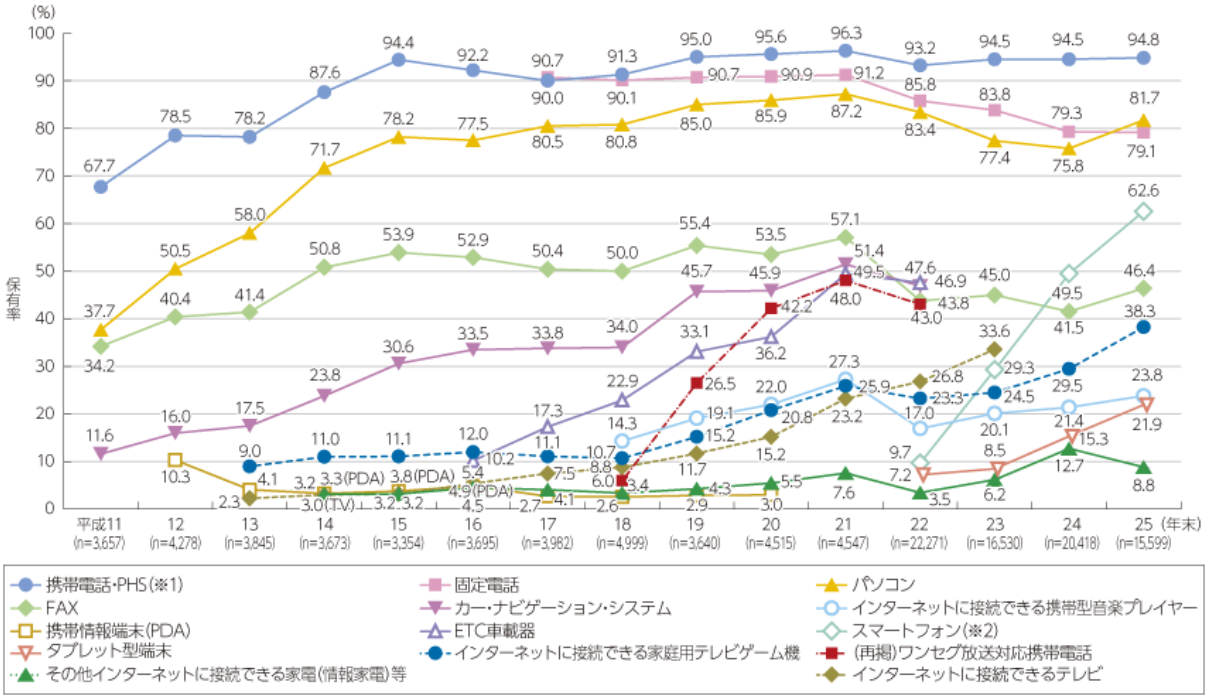

図 2.5: 情報通信端末世帯保有率の推移 (出典:[22])

体的には AndroidOS [1] ,iOS[2] などを搭載しているものである。またタブレットとはスマートフォンと同じ要件を満たしながら、画面の大きさがスマートフォンよりより大きい物を意味している。

2.2.2 モバイルデバイスの用途

ユーザーはスマートフォンを使うことでパソコンと同様に多くのことができる。その中でもスマートフォンユーザーは文字入力を行う可能性が高いアプリケーションを頻繁に使用している。リサーチバンクによるアンケート調査 [6] (図:2.6) 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 アプリはそれぞれスマートフォンを情報の受信端末としてのみ使う場合には文字入力の必要はないが、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。スマートフォンにおける他のアプリケーションにおいても必要に応じて文字入力を行う。こういった結果から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も多いことがわかる。

[1] http://www.google.com/intl/ja/mobile/android/
[2] http://www.apple.com/jp/ios/

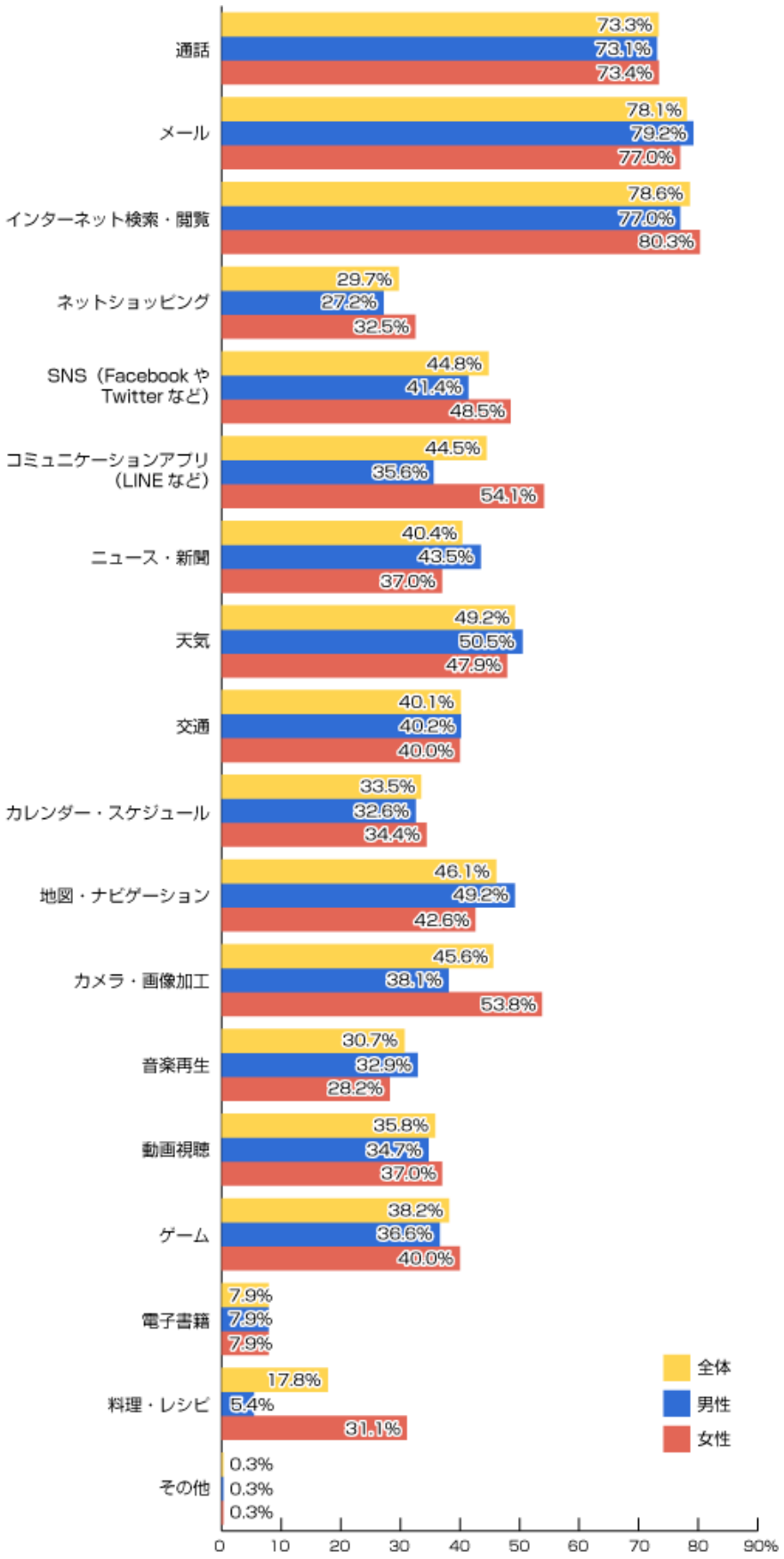

図 2.6: スマートフォンでどのような機能・アプリを使っているか (出典:[6])

2.2.3　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:2.7) や、ヘッドマウントディスプレイの一つである Google Glass(図:2.8) がその例として挙げられる。ス

図 2.7: スマートウォッチの例:Android Wear を搭載した Moto360(出典:[1])

図 2.8: Google Glass(出典:[2])

マートウォッチは既存の時計にモバイルデバイス向け OS を搭載したもので、従来の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。Google glass においても同様でメガネにおいて必要のなかった文字入力がこういったデバイスになることで文字入力が必要となる。

2.3　日本語入力と英語入力の差異

日本語と英語ではコンピューターに対する入力にかかるコストが大きく異なる。qwerty キーボードで日本語を入力する場合、アルファベットをキーボードで順番に打ち、打ち込まれたアルファベットをにひらがなに日本語 IME が変換する。そして変換されたひらがなを日本語 IME は漢字やカタカナの候補に変換しユーザーに提示する。ユーザーは提示された変換候補などを選択することで文章を完成させるというプロセスになる。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 を使う必要がなく、ほとんど使われていないのが現状である。このため英語圏においては研究が盛んではなく、世界中での IME やそれに対する研究の需要は増している。

2.4　入力システムの変遷

今日ではパソコンに遜色のないような CPU やメモリを積んでいるスマートフォンも多く市販されている。これらのデバイスの性能は携帯電話の頃から考えると大幅に向上している。デバイスの性能が向上していく間に IME にできることが増え、それと共に様々なテキスト入力方式などが考案された [23] ものの搭載されている入力方式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システム [7] が開発され、IME への導入がおこなわれた。これらのシステムは様々に応用可能で広く普及したがいまだ日本語 IME への改善の余地は多く残されている。

第3章　推薦システム

本章ではコンテキストを用いた候補単語推薦システムについて述べる。

3.1 概要

本推薦システムはユーザーのコンテキストと入力を受け取り、過去の全てのユーザーの入力情報と比べることによって入力する単語を予測し推薦するものである。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。この一連のプロセスを繰り返すことでユーザのコンテキストに適した候補単語の推薦を行う。本論文においては推薦候補単語とは本推薦システムを通してでた結果から取得する候補単語のことであり、一般の仮名漢字変換や予測入力における候補単語は含まない。

3.2 取得コンテキスト

ここでは推薦システムにおいて使用しているコンテキストについて述べる。コンテキストを何度も取得することで本システムは成り立っているが、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大きな負荷がかかるため、入力開始から終了までコンテキストが変化するものと変化しないないものに分類し、コンテキストを取得するクライアントを実装することで負荷の軽減に役立てた。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画面において設定するものと、Rive 日本語入力が自動的に取得するものがある。

3.2.1 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アプリケーションことである。AndroidOS においては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の推薦に役立てる。

(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の推薦に役立てる。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、同じ年齢あるいは近い年齢の人が使っている単語の推薦に役立てる。

(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の推薦に役立てる。クラスタとは同様の嗜好を持った人々をまとめたグループのことである。キャラクターの要素はユーザーが以下から選択する。

- ヲタク
- JK
- 熱血
- 冷静
- おねぇ
- 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の推薦に役立てる。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取得しないようにしている。

3.2.2 動的コンテキスト

以下動的コンテキストとして利用しているものについて述べる。

(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の推薦に役立てる。現状では時間と分を取得し、秒以下の数値は切り捨てている。

(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の推薦に役立てる。

(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の推薦に役立てる。

(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の推薦に役立てる。

(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に役立てる。

(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、共起表現のような入力単語と同じ文章内で使われやすい単語の推薦に役立てる。

(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うち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、アプリケーションに入力を開始した直後に入力しやすい単語や、文章の区切りにおいて入力しやすい単語などの推薦に役立てる。

3.3 Jubatus

3.3.1 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。Rive 日本語入力ではコンテキストを常に取得し計算するため高速な処理が必要となり、高速かつスケーラビリティに優れていた[18]、Jubatus を採用した。

3.3.2 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けし、推薦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。

3.4 推薦例

本システムにおいて起こる推薦の具体的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
  LINE 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。

第4章　設計と実装

本章ではRive日本語入力全体の設計と、構成する各システムについての詳細な実装の解説を行う。

4.1 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
- DataBase

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を実現している。(システム全体構成図:4.1)

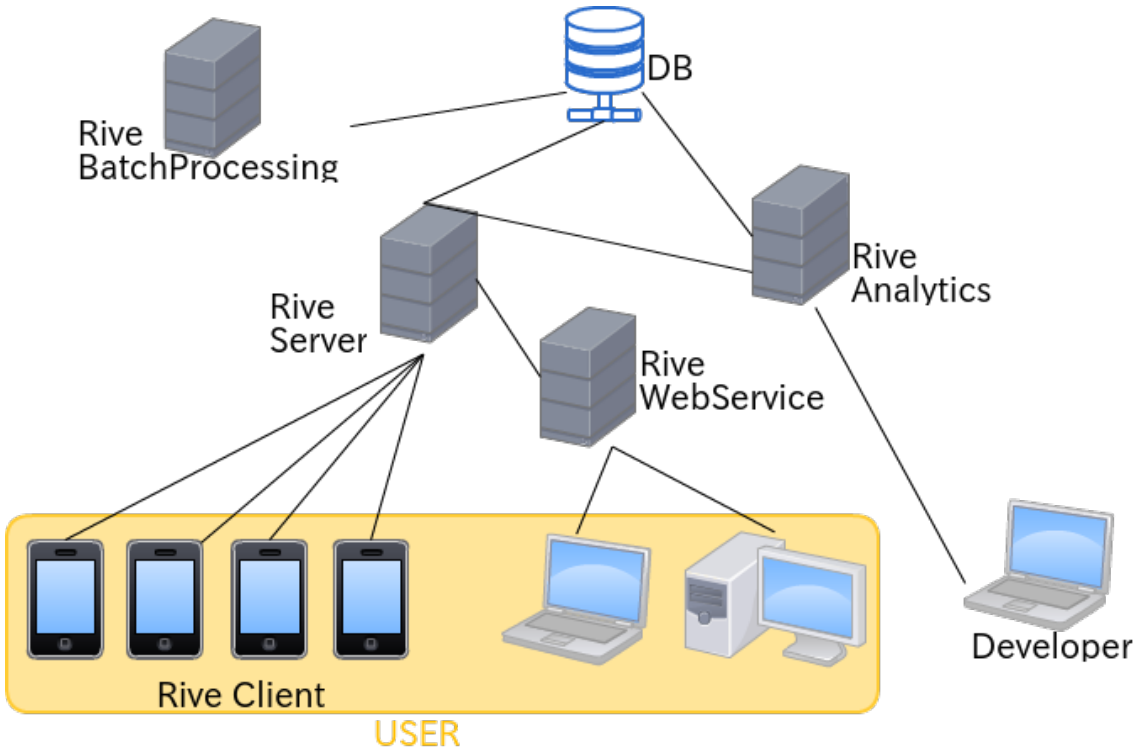

図 4.1: システム全体構成図

4.2 Rive Client

本システムは AndroidOS 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可能である。(Rive Client ユーザ画面:4.2)

図 4.2: Rive Client ユーザー画面

4.2.1 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() メソッドが呼ばれ、システムのイニシャライズを行う。その後 onStartInput() メソッドが呼ばれコンテキストを取得する。ここで取得するコンテキストは静的コンテキストと動的コンテキストの両方である。コンテキストを取得し次第、この候補単語はサーバーと通信した上で推薦候補単語を取得する。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である onKeyDown() メソッドや、候補の単語をタッチする pickCandidateWord() メソッド、あるいは文字をデリートする deleteWord() メソッドも含まれる。二回目移行のコンテキスト取得においては動的コンテキストのみの取得となる。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:4.3)

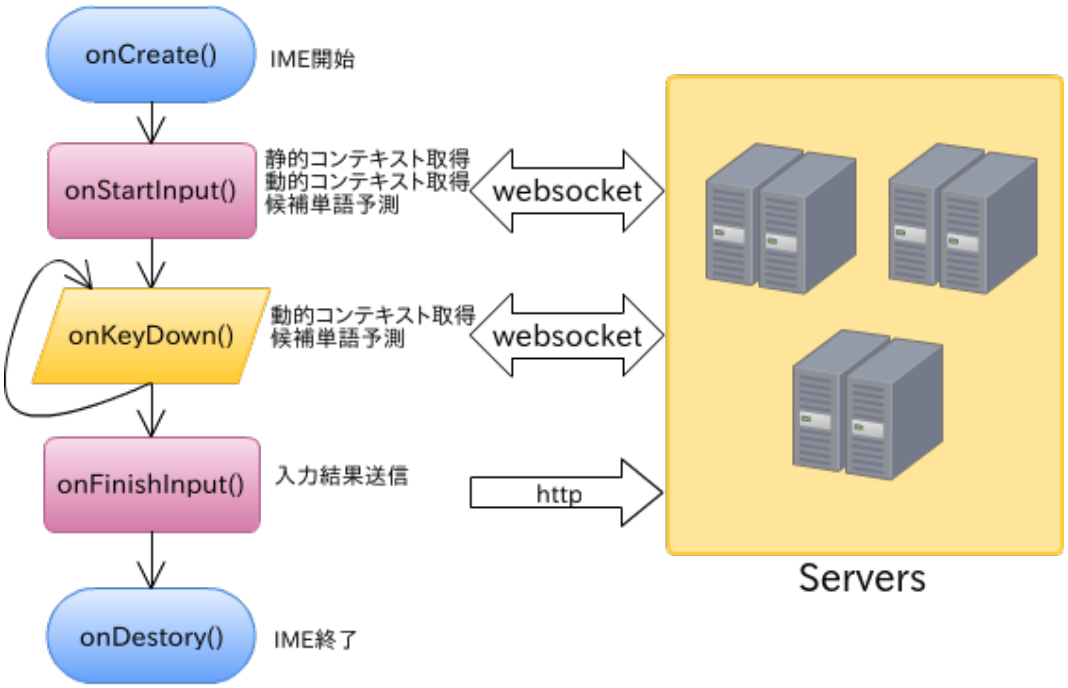

図 4.3: Rive Client フローイメージ

4.2.2 コンテキスト取得

取得するコンテキストについては 3.2 項を参照。物理的なセンシングからは推測が困難であるが、有用であると考えられるコンテキストは設定画面においてユーザーが入力を行う。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。

4.3 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郡の総称である。Rive Client からコンテキストデータを受け取り、集合知を用いて解析を行う。解析し適した候補単語を推測し、それら候補単語を Rive Client に送信する。また Rive Analytics に入力データを送信するという役割も担っている。

4.3.1 システム構成

サーバーの中身は大きく Routting Server,beforeRules,afterRules,Jubatus の 4 つから成り立っている。Routing Server は非同期処理に適した Node.js[1] 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:4.4) ような分割になっている。始めにクライアントが

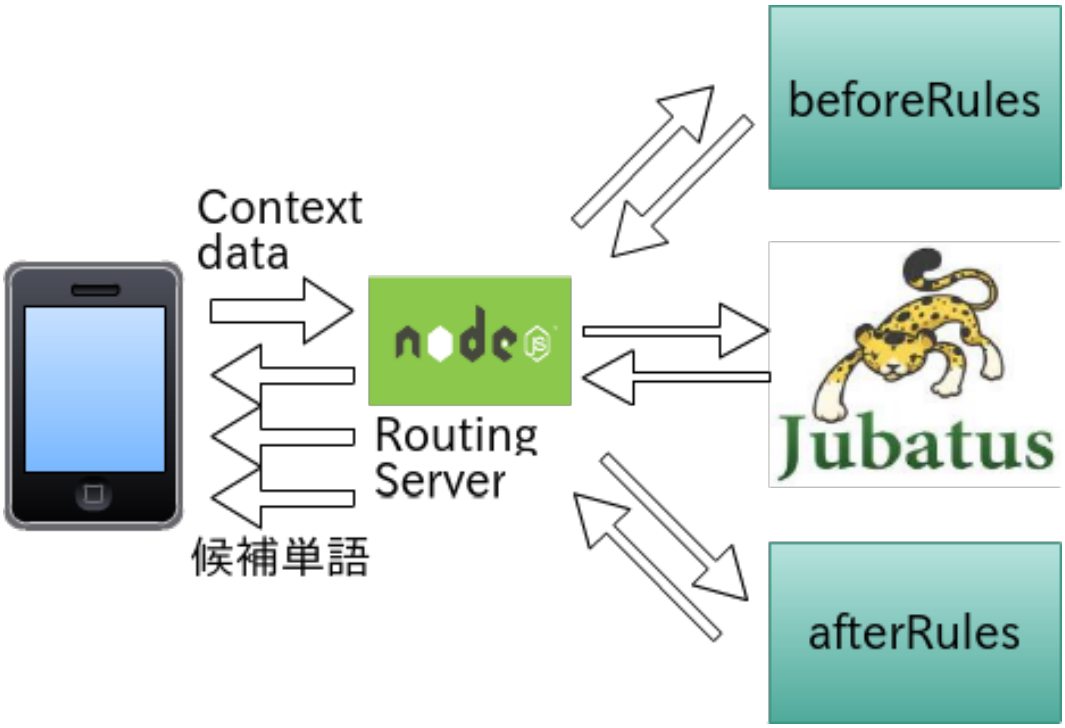

図 4.4: Rive Server 概要図

コンテキストデータを Routing 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ントに返すことで完了する。一回

[1] http://nodejs.jp/

のタッチごとにこれら全てのプロセスを行う。クライアントは非同期に受け取った推薦候補単語をスコア順にソートして表示する。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。

4.3.2 Routing Server

Routing Server は二通りの役割を担っている。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 つの計算エンジンのどれに振り分けるかを判断し、それぞれに送信する役割である。もう一つはデータを受け取り次第、デバイスに返信する役割である。

4.3.3 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推薦候補を計算す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、Twitter[2] 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。

4.3.4 Jubatus

このエンジンについては 3.3 項参照。

4.3.5 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っ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という処理などである。

4.4 Rive Analytics

このシステムは Rive Client でのプロセスの終わりに送られてくる入力データを受け取り、解析するシステムである。開発者はこのシステムを開発時に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。バージョン管理システムである git[3] の commit[4] とデータを紐付けて入力データを管理する。ユーザーは直接本システムとの関わりは持たない。

[2] http://twitter.com/
[3] http://git-scm.com
[4] http://git-scm.com/docs/git-commit

4.4.1 指標値に用いる単語の説明

click デバイスにおける全てのタッチ

touch 候補単語のタッチ

before beforeRules(4.3.3 項) からのタッチ

after afterRules(4.3.5 項) からのタッチ

jubatus Jubatus(3.3 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

4.4.2 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省入力化を達成できる。
- 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 - Before rules 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 - After rules 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:4.5) 必要な指標値と開発におけるバージョンを見比べながら有効であると思われる機能

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 4.5: Rive Analytics 画面

についての開発の参考にする。

4.5 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。

4.6 Rive Webservice

このシステムはインターネット上で利用可能な Web ページとして実装した。Rive Server の推薦を試用することができ、また Rive 日本語入力の使い方などを説明している。[5] Rive Server と通信を行い、Rive Webservice 上で仮想のコンテキストを設定することで、どのような推薦候補単語を受け取ることができるかシミュレーションすることができる。(体験画面イメージ:4.6)

[5] http://rive.in

図 4.6: Rive Webservice の体験画面

4.7 データベース

取得するコンテキスト共に、スキーマが頻繁に変わるためスキーマの変更に対応しやすい NoSQL データベースである、MongoDB[6] を採用した。

4.8 システム間通信

Rive Client と Rive Server 間の通信は Websocket[7]、その他のシステム間は 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 より速度が早いためという理由である [3]。

[6]http://www.mongodb.org/
[7]http://dev.w3.org/html5/websocket/

第5章　ユーザインタフェース

本章では Rive 日本語入力におけるユーザインタフェースについて述べる。

5.1 概要

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 を切り替えたりせず、あらゆるアプリケーションに対して同じ IME を利用する。そのため Rive 日本語入力では基本的な入力をすることができ、かつ推薦機構の導入を実現しる必要がある。そこで IME 本来の機能を損なわないように独自のインタフェースを実装することで Rive Client における基本的な入力と推薦機構を両立した。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 への適応が可能である。

5.2 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:5.1, 図:5.2)

図 5.1: qwerty キーボード

図 5.2: flick キーボード

5.3 ダブル辞書インタフェース

IME としての通常の文字変換を併用しないと普段の使用において支障が出るため、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 に搭載されている変換システムである。それぞれ横方向にスクロールさせることで候補を順番に見ることができる。(ダブル辞書インタフェース画面:5.3)

5.4 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、新たな候補単語郡を推薦する。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

図 5.3: ダブル辞書インタフェース画面

ク操作では類語変換を行う。この機能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。この機能は連続で使用することができるため、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:5.4)

5.5 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する際に、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。

図 5.4: 単語フリックイメージ図

第6章　課題

本章ではRive日本語入力における課題とそれに対するアプローチについて述べる。

6.1 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、データが充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がある。本来こういった集合知を使ったシステムにおいては、外部システムやアプリケーションから持ってきたデータ等をシステム用に加工し、シード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である。そこで第 4.3 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。

6.2 省入力化と多様性のジレンマ

Rive 日本語入力を使用し、キーボードを使うことなく候補単語の選択だけで入力が完了するようになると本システムの目的である省入力化は達成される。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ある。これ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誤解から起こる問題であると考える。物理的なコンテキストしか取得していないと確かに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。物理的なコンテキストとは気温や位置などの客観的に測ることができるコンテキストのことであり、精神的なコンテキストとはユーザーが怒っているとか悲しんでいるなどといった、客観的な評価が難しいコンテキストのことである。精神的なコンテキストを集め、精神的状況をきちんと把握することができれば多様性を持った充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ていく。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、いずれはカメラによるユーザーの表情解析や、ユーザーの体温等の物理的センシングから得られる情報を使って精神的コンテキストを推測し、入力の多様化を担保したいと考えている。また単語変換機能 (第 5.4 項) における類語変換や英語変換機能もユーザーの入力の多様性の向上に役立っている。

6.3 プライバシーの問題

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で、推薦候補を推測しているがこれはプライバシーという観点から見た時に問題になりえるのではないかということである。

6.3.1 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扱っているデータ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対応している。

6.3.2 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。しかし入力データの送信は Rive 日本語入力において必要不可欠な要素であり、現状ではユーザーに事前にデータ送信の要請をすると共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7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。

7.1 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。

7.1.1 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

荒川らが研究・提案した動的辞書生成手法 [20] である。ユーザのコンテキストに応じて適切な単語を推薦することによって、携帯端末の文字入力を改善するコンテキストアウェア IME の概念を提案し、そのために必要な動的辞書生成手法に提示している研究である。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定義した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。

7.1.2 Social IME

奥野が開発したユーザ間で辞書を共有する IME[17] である。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 [16] 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。

7.1.3 iWnn

iWnn とは [9] omron 社が開発した日本語 IME であり、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。

7.1.4 Google 日本語入力

Google 日本語入力は Google 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ることができるという点に置いて Rive 日本語入力と類似している。その上でリアルタイムな情報を推薦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。

7.2 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発展してきた [15]。

7.2.1 コンテキストの分類

Rive 日本語入力においては第 3 章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、コンテキストには様々な分類方法が存在する。まずソフトウェア開発に関するコンテキストでは Kai Peterson らによる研究で [10] Product,Process,Practice/Tool,People,Organization,Market という 6 つにコンテキストが分類されている。しかし Rive 日本語入力において想定しているコンテキストとは多少異なるため、Rive 日本語入力では主に Strang らの研究 [12] を参考にした。その研究によるとコンテキストとは以下に分類される。

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

Rive 日本語入力においては集合知を用いた機械学習手法により、コンテキストを用いた推薦を行うため、コンテキストを決める際の参考ではなく特に beforeRules や afterRules 等の閾値を考察する際の参考とした。

7.2.2 コンテキストを用いた入力推薦システム

ここでは Rive 日本語入力を作る上で参考とした、コンテキストを用いた推薦システムについて述べる。特に Rive 日本語入力に近いと思われるものについて挙げる。まず Rohodes らによるシステム [11] が挙げられる。このシステムはウェアラブルデバイスを使ってコンテキストを取得し、emacs[1] における文字入力を助けるシステムである。このシステムにおいては 5 秒ごとにコンテキストを取得し、ユーザーにエディタにおける最適な文字入力の提案を行うというシステムである。エディタという限られた状況に絞っているためこのシステム

[1] http://www.gnu.org/software/emacs/emacs.html

は有用であるが、IME では様々な状況で様々な入力を行うため、予め開発者によって決められている候補だけでは充分ではない。Rive 日本語入力においては過去の全てのユーザの入力を使うことでこの問題を解決している。また高橋らによる状況と行動に関する研究 [21] も挙げられる。この研究ではコンテキストにおけるユーザの思考を行動という形で表しているが、これは Rive 日本語入力における候補単語を選択するという入力の方式に類似している。また同様な研究にはグェンらによる研究 [14] もある。しかしこれらのシステムは限定的な状況におけるコンテキストによる予測を行っており、あらゆる状況における入力を想定している Rive 日本語入力とは異なるが限定的な状況においてより有用な推薦をするための参考になる研究といえる。

第8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。

8.1 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。

8.2 システムへの評価

Rive 日本語入力は公式での配布は行っていないが同じ研究室に所属している人や友人などに配布し実際に使用してもらうことで、

- 充分に使用可能な速度になっていること。
  IME において使用可能な速度になるように最新技術をふんだんに使うことで達成できた。
- 推薦候補が来るとローカル環境においてはすごく面白いと共に恐怖であるということ。

  少ない人数で運用していたため、身近な人々が使っている単語などが推薦され、推薦候補単語はとてもおもしろいものが得られたが、それと共に誰がどんな入力をしたのかがすぐにわかってしまい、プライベートな入力はとてもしづらい状況になってしまった。
- 候補単語変換機能は英語があまり得意でない人にはとても役に立つということ。
  多様性を確保するために考案した候補単語変換機能であるが、入力に悩んだ時などにとても効果を発揮し、楽しい入力を行うことができた。
- コンテキストとして最も有用であるのはアクテビティーであるということ。
  ユーザーは入力するアプリケーションごとに、文体や語句を使い分けることが多く、アクテビティーによるコンテキストを用いた推薦が最も効果を発揮しているということがわかった。

といった結果を得ることができた。

8.3 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のである。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検討している。またデバイスが新しいコンテキストの取得が可能になれば、推薦の精度が多少なりとも向上するため、システ

ムの有用性はデバイスの進化と共に上がると考えている。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。

8.4 研究への考察

ここでは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を行う。最新の技術を利用することで、旧来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、提示するために充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9章　本研究に関連する発表

9.1　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)

9.2　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: コンテク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 期スーパークリエーター達 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いてくれた両親、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] Kai Peterson and Claes Wohlin. Context in industrial software engineering research. *IEEE*, 2009.

[11] B. J. ROHODES. The wearable remembrance agent : A system for augmented memory. *Personal Technologies*, Vol. 1, No. 4, pp. 218–224, 1997.

[12] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. *Ubicomp*, 2004.

[13] Jenifer Tidwell. デザイニング・インターフェース 第 2 版. O'REILLY, 2011.

[14] グェンミン テイ, 隆浩川村, 博之中川, 康之田原, 昭彦大須賀. Web からの自己教師あり学習を用いた人間行動マイニング (一般,「グリーン ai」及び一般). 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 109, No. 386, pp. 19–24, jan 2010.

[15] 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.

[16] 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.

[17] 奥野陽. Social IME. http://wwww.social-ime.com.

[18] 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.

[19] 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.

[20] 荒川豊, 末松真司, 田頭茂明, 福田晃. 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. 情報処理学会論文誌, Vol. 52, No. 3, pp. 1033–1044, 3 2011.

[21] 高橋公海, 佐藤進也, 松尾真人. 状況に依存した行動パターン抽出手法の検討. 情報処理学会研究報告. 情報学基礎研究会報告, Vol. 2013, No. 24, pp. 1–6, jul 2013.

[22] 総務省. 平成 25 年通信利用動向調査. Technical report, 総務省, 2012.

[23] 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.