

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 豊^{†1} 末松 慎 司^{†2}
田頭 茂 明^{†1} 福田 晃^{†1}

本論文では、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、携帯端末における日本語入力を改善するコンテキストウェア IME の概念を提案し、それを実現するために必要な動的な辞書生成手法について述べる。我々は代表的なコンテキストとしてユーザの現在位置に着目し、現在位置に基づいた動的な辞書作成手法、十分な応答速度を満たすシステムアーキテクチャ、Web 検索ヒット数に基づくソート手法を提案する。そして、提案手法を PC および Android 端末上に実装し、十分な応答速度で辞書を更新可能であることから、提案システムの有効性を示す。

Dynamic Dictionary Generation Method for Context-aware Input Method Editor

YUTAKA ARAKAWA,^{†1} SHINJI SUEMATSU,^{†2}
SHIGEAKI TAGASHIRA^{†1} and AKIRA FUKUDA^{†1}

In this paper, we propose a concept of context-aware IME (Input Method Editor) for improving the input of Japanese on mobile devices. In our concept, according to the user's current location, a personal context-aware dictionary is dynamically generated from the keywords gotten via some APIs in the Internet. We propose dynamic dictionary generation method, system architecture, sorting algorithm based on a result of web search. The result of response time of our proto type systems show its effectiveness.

^{†1} 九州大学大学院システム情報科学研究院

Graduate School of Information Science and Electrical Engineering, Kyushu University

^{†2} 九州大学大学院システム情報科学府

Faculty of Information Science and Electrical Engineering, Kyushu University

1. はじめに

近年、携帯端末の高機能化や iPhone や Android 携帯に代表されるスマートフォンの台頭とともに、従来 PC で行ってきた作業が PC 上から携帯端末上へとゆるやかにシフトしている。メールの確認や送受信といった作業はその代表例である。また近年の調査¹⁾では、インターネットアクセスの5割以上が携帯端末からのアクセスであることや、8割以上のユーザが携帯公式サイトなどの階層化されたメニュー経由ではなく、パソコンと同様に Google などの検索エンジン経由で欲しい情報にアクセスしていることが判明している。その結果、携帯端末における文字入力への頻度は増加の一途をたどっており、省入力化への要求が高まっている。

省入力化の代表的なアプローチとして、1) キー入力方式の改善、2) 辞書の拡充、3) 予測変換、4) 学習、などがあげられる。たとえば、1) の例としては、一般的な入力方式であるマルチタップ方式(押す回数で「あいう」と変化する)に対して、子音と母音のタッチで入力するボケベル方式、入力したい文字が割り当てられているキーを1回だけ押し文字列を推測するシングルタップ方式 T9²⁾、入力したい文字が割り当てられているキーを押し、その状態から指を四方にスライドさせることによって入力するフリック方式などがある。2) の例としては、ダウンロード辞書機能があげられる。これはインターネット用語辞典や人名辞典など、特定の分野に特化した辞書を目的に応じて追加できる機能である。さらに近年では、ネットワークで辞書を共有し、利用者全員が単語の登録・共有を行うことのできる Social IME³⁾が提案されている。3) の例としては、前方一致検索による全体文字列の推測や、文脈に基づいた助詞・助動詞などの推測があり、PObox⁴⁾を筆頭に、現在広く普及している。4) の例としては、仮名漢字変換や予測変換での単語候補において、使用頻度と使用履歴に基づいたソートが行われるのが一般的である。特に学習を用いた予測変換は、個人の嗜好を反映しているため、メールの作成などの日常的な文字入力シーンに対して有効なアプローチとなっている。

しかしながら文字入力のシーンは多様化してきており、メールやメモの作成だけでなく、乗り換え案内の利用や周辺情報の検索なども携帯端末上で行うようになってきた。このような多様な文字入力シーンに対しては、使用頻度や使用履歴という指標の一律な適用だけでは十分に対応することができない。そこで近年では、ユーザの状態(コンテキスト)を推定し、より入力時の状態に即した単語を推薦する研究が行われている。携帯端末向け IME である iWnn⁵⁾では、電話帳の登録情報や季節・時間帯などを利用し、予測変換候補を動的に変化

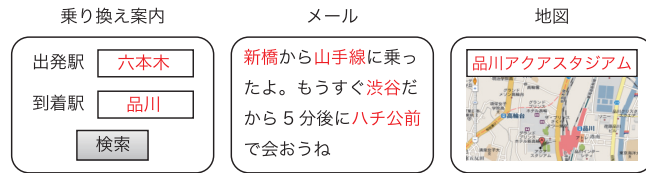


図1 提案するコンテキスト IME の利用例
Fig. 1 Suitable situations for context-aware IME.

させることで省入力化を支援する仕組みが実装されている。しかしながら、利用しているコンテキストは端末上で取得可能な情報（ローカルコンテキストと定義）に限られており、モバイルコンピューティング環境において特徴的なコンテキストである位置情報が考慮されていないなど、改善の余地も多い。また、これらの手法を用いた場合も、初めて入力する地名やニッチなランドマーク名（たとえば、ビル名や交差点の名前、レストラン名）など、辞書データに登録されていない文字列に関しては変換候補として提示することはできない。

そこで本論文では、携帯端末における新たな省入力化へのアプローチとして、より適した文字列を推薦するために、ユーザのコンテキストに応じて動的な辞書を生成するコンテキストウェア IME システムを提案する。提案システムは、入力インタフェースや過去履歴学習、文脈解析など、既存 IME が備える機能はそのまま利用することを想定し、内部辞書の動的な更新に主眼を置いている。これは、近年の IME はきわめて高度化しており、全体的な性能や提案システムの普及の観点から、それらと共存できる形態が望ましいと考えたからである。提案システムでは、位置情報やネットワークを介して得られるプレゼンス情報（グローバルコンテキストと定義）、さらにそれらから副次的に得られる周辺情報（ランドマーク名、最寄り駅名、レストラン名）などを考慮し、ユーザのいる場所・時間・状態に応じて、動的に辞書を生成する。提案するコンテキスト IME の利用例（図1）としては、乗り換え案内を利用する際に、出発駅として最寄り駅が優先的に推薦されたり、目的駅をスケジュール情報から推定したりすることが考えられる。ほかに、その時間にテレビに出演している芸能人の名前が予測変換候補として優先的に表示されたり、旅行にいくと、その場所付近の観光名所がすでに辞書に入っていたり、「し」で始まる「新宿」「渋谷」「新橋」などが位置によりソートされていたりといった効果を狙っている。それを実現するための手法として、携帯端末に搭載された GPS センサや加速度センサ、地磁気センサなどから、ユーザの位置や移動方向を取得するとともに、ネットワークを通じて周辺情報やスケジュール情報、プレゼンス情報などを取得し、ユーザのコンテキストを推定する。次に、推定したコンテキスト

に基づいて、動的に辞書を作成する。辞書の作成方法としては、ネットワーク上に公開されているさまざまな Web API を用い、単語の取得および読み仮名の付与を行う。通常、ある辞書を作る際には初期コストやメンテナンスコストなどの膨大な人的コストが必要とされるが、Web API から提供されるデータを活用することで、コスト負担なしに辞書作成を行うことができる。さらに、すべての Web API を辞書全体と見なせば、辞書内の単語は Web API の種類によりクラスタリングされており、Web API へのクエリによってフィルタリング可能であるといえるため、詳細かつ的確な単語が取得可能であると考えられる。このようにして作成された辞書内の単語をもとに、予測変換候補としてユーザに提示することで、状況に応じた単語の推薦を実現する。さらに、単語をソートする手法として、インターネット上でよく用いられている単語ほど、重要度が高いと考え、Web 検索のヒット数を用いたソート手法を提案する。また、本システムでは、サーバ側で辞書を生成するため、その応答時間が重要となることから、本システムを実現するためのシステムアーキテクチャについても検討を行い、十分な応答速度が得られるアーキテクチャを提案する。

本論文の構成を以下に示す。まず、2 章で関連研究について述べ、4 章で提案システム、5 章で実装したプロトタイプについて説明する。6 章で提案システムの有効性を検証し、7 章で総括する。

2. 関連研究

本論文に関連する研究分野として、コンテキスト情報を活用した入力単語予測について述べる。さらに、要素技術として必要である、日本語入力ソフトに置ける辞書拡充、複数のセンサ情報からのコンテキスト推定、マッシュアップサービスについて述べる。

2.1 コンテキスト情報を活用した入力単語予測

ユーザのコンテキストに基づく入力単語の予測に関する研究は数多く行われている。服部ら⁶⁾は、ユーザの地理的な位置や周囲にあるモノ群に関する情報を活用することで、コンテキスト・ウェアな検索質問の修正および検索キーワードの推薦手法について提案している。提案手法では、あらかじめある単語とコンテキスト語との関連度を検索サイトの検索数の変化から算出しておき、ユーザのコンテキストに応じて動的な検索キーワードの推薦を行っている。たとえば「は」で始まる文字列について、ハリーポッターならば書店、ハリガネムシならば学校という場所語に関連付けておき、ユーザの現在地に応じて推薦キーワードを変化させている。また、中村⁷⁾は、web 上での文章作成補助を目的とし、文書中の単語から必要とされる属性値を取得し（たとえば、あるランドマーク名からその住所を取得）、

ユーザに提示する手法を提案している。しかし、これらはいずれもある web ページ上の特定の入力ボックスにおいて動作するものであり、文字入力シーン全般に対して汎用的に動作するものではない。

日本語入力ソフトに関してコンテキストの利用を行っているものとしては、前述した iWnn がある。これはローカルで得られるコンテキスト情報を用いることで、変換候補を動的に変化させている。たとえば、時間を判断して朝と夜で「おはよう」「おやすみ」の優先度を変える、電話帳からメールの送信相手のプロフィールを判断し、口語表現と敬語表現の優先度を変えるなどを行っている。また、小松ら⁸⁾ はコミュニケーションシーンに着目し、メールやチャットの受信文に含まれる単語を優先的に表示するといった、文脈をとらえた予測変換手法を提案している。

また、Kukura⁹⁾ では対象とするコンテキスト情報をさらに広げ、ユーザが観覧中のウェブページなどの情報も取り入れている。Kukura 自体は文書蓄積システムとして振る舞い、蓄積した文書から単語やその頻度を抽出し、POBox などの予測入力システムへ提供している。これらのシステムを用いることで、ユーザは過去に入力したことのない単語であっても予測変換候補として推薦できるという利点がある。

2.2 日本語入力ソフトにおける辞書拡充

MS-IME や ATOK などは、インターネット用語辞典や人名辞典など、特定の分野に特化した辞書を目的に応じて追加できる機能を有している。また、Google IME や Baidu Type は Web 上の膨大な単語を辞書に格納することで、新語や流行語といった時事的な語彙が豊富なほか、くだけた表現・口語表現も辞書に有している。一方、こうしたローカル辞書の拡充とは別に、奥野ら³⁾ は、辞書データをインターネット上に置き、ユーザの変換結果や辞書登録単語を集合知として蓄積して共有することで予測変換候補のバリエーションを増やす Social IME を提案している。この方式も Google IME などと同様に新語や流行語をいち早く辞書に反映できるというメリットを有している。我々の提案システムも Social IME と同様にインターネット上に辞書を置き、動的に更新していく。

2.3 コンテキスト推定

我々の提案システムはコンテキストウェアサービス的一种であるが、種々のセンサ情報からユーザのコンテキストを推定し、サービスへ応用する研究は数多く行われている。猿田ら¹⁰⁾ は加速度センサや RFID から得られたセンサ情報から特徴量を算出し、「座る」、「立つ」、「書く」、「飲む」などの日常的な動作を推定する手法を提案している。森岡ら¹¹⁾ も同種の手法により、「歩く」、「走る」といった動作や階段やエレベータの昇降の判別を行って

いる。またコンテキストウェアなサービスとしては、ユーザの好みと現在のコンテキストを考慮したプッシュ型のコンテンツ配信サービス¹²⁾ や、ルールベースの行動支援サービス¹³⁾ などがある。上記のサービスはそれぞれ独自のコンテキスト推定エンジンをもとに実現されているが、より柔軟性を持たせることを目的として、コンテキスト情報の取得を統一的に扱うミドルウェアも提案されている^{14),15)}。さらに、川原ら¹⁶⁾ の提案するコンテキストウェアサービスプラットフォーム “Synapse” では、センサ情報とサービスとの関連を確率的に学習し、好みや習慣を反映したサービスを自動生成する手法を提案している。また、行動ログをベイジアンネットワークで分析することによりコンテキストを推定する手法¹⁷⁾ や、位置情報をもとにユーザの滞在するエリアを識別し、会社員、学生、主婦などのユーザの属性ごとに、事前に定義した行動の遷移パターンを用いて高精度にユーザの行動を推定する方式¹⁸⁾ なども研究されている。コンテキスト推定に関しては、これらの従来技術を利用することを想定し、本論文では取り扱わない。

2.4 マッシュアップサービス

現在、複数の Web サービスを連携し、新たな 1 つの Web サービスとして提供することがさかんに行われている。これをマッシュアップと呼び、さまざまな Web サービスが連携に必要なインタフェースを API として一般のユーザへ公開している。Web API の利用に際しては、SOAP や REST といったプロトコル、XML を使ったデータのやりとりを行うことで、開発者側の負担が軽減されている。公開されている Web API の種類は地図や天気、駅名、商品名の情報提供など多岐にわたる。マッシュアップサービスの例としては、地理情報に対応したコンテンツを地図上に表示させるものや¹⁹⁾、あるキーワードについて議論しているブログとショッピングサイトの関連製品を同時に表示させるものなどさまざまである。我々の提案システムでは、このマッシュアップ技術を用いることにより、動的な辞書の生成を実現している。

3. 予備実験

ここでは、位置というコンテキストと実際に入力される文字列に関係性があるかを確認するための予備実験について述べる。この関係性を調べるためには、一般的には被験者に移動してもらい、さまざまな位置で文字入力を行ってもらう必要がある。しかしながら、端末数や人的制約から被験者を用いた検証には限界がある。そこで、位置と入力文字列の関係性を明らかにする手法として、インターネット上で取得可能な位置情報付き文字列、具体的には Twitter (ツイッター)²⁰⁾ を分析する方法を提案している²¹⁾。Twitter とは、2006 年 7 月

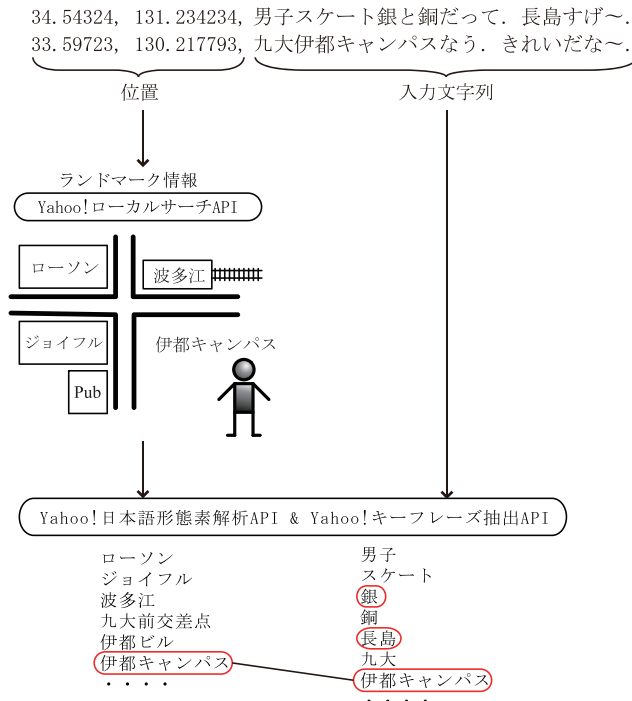


図2 ツイートから得られる情報とその分析の流れ
Fig. 2 How to analyze the information obtainable from tweets.

に Obvious 社が開始した，ユーザが 140 文字以内で「つぶやき（ツイート）」を投稿することで，メールやメッセンジャよりも，ゆるいつながりを発生させるコミュニケーションサービスである．ツイートには GPS から取得した位置情報を Geo タグとして付与することが可能であり，API 経由でツイートを取得できる．図 2 に示すように，1 つのツイートから，つぶやいた位置，つぶやいた文字列，という情報を得ることができる．そこで，位置情報をもとに Yahoo! ローカルサーチ API から得られる単語と，実際のツイートに含まれる単語を分析することで，位置と入力文字列の関係を明らかにできると考えた．

結果としては，ツイートを発した位置にもとに Yahoo! ローカルサーチ API から得られるランドマーク情報を，ツイート自身に含んでいた割合（含有率）は，4.83%（13,590 件中 656 件）であった．この数字は，少ないようではあるが，20 ツイートに 1 ツイートは，周

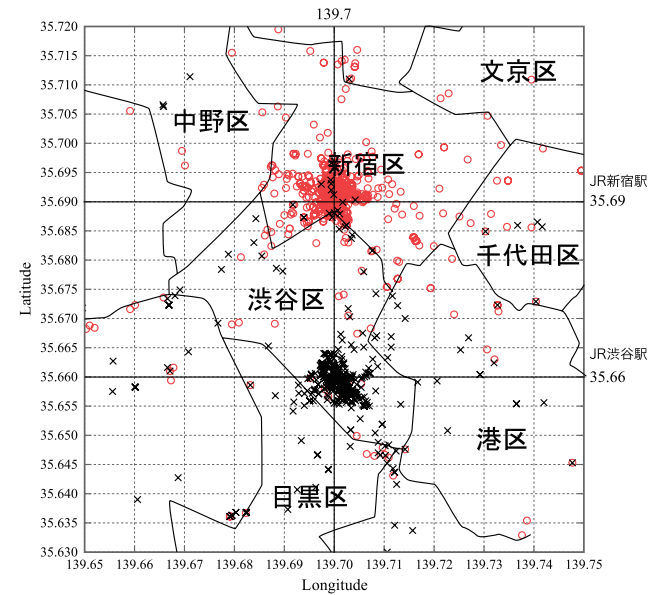


図3 「新宿」と「渋谷」を含むツイートの地理分布
Fig. 3 Geographical distribution of tweets including “Shibuya” and “Shinjuku”.

辺のランドマーク情報を含んでいることは事実であり，検索候補の絞り込みアルゴリズム次第では有用であると考えられる．図 3 は，「新宿」を含むツイートと「渋谷」を含むツイートの地理分布を示している．図より，「新宿」という文字列は新宿エリアに集中し，「渋谷」という文字列は渋谷エリアに集中していることが明らかであり，これらの単語は位置に依存しているといえる．「新宿」も「渋谷」もともに，「し」で始まる単語であるため，この結果から，入力文字「し」に対する変換候補を位置に応じて変化させることは有効であると考えられる．

4. 提案システム

本論文では，ユーザのコンテキストに応じて辞書データをダイナミックに更新するコンテキストウェア日本語入力支援システムを実現するために，1) 十分な応答速度を実現するシステムアーキテクチャ，2) 位置に応じた動的な辞書生成手法，3) Web 検索ヒット数に基づいたソート手法，を提案する．

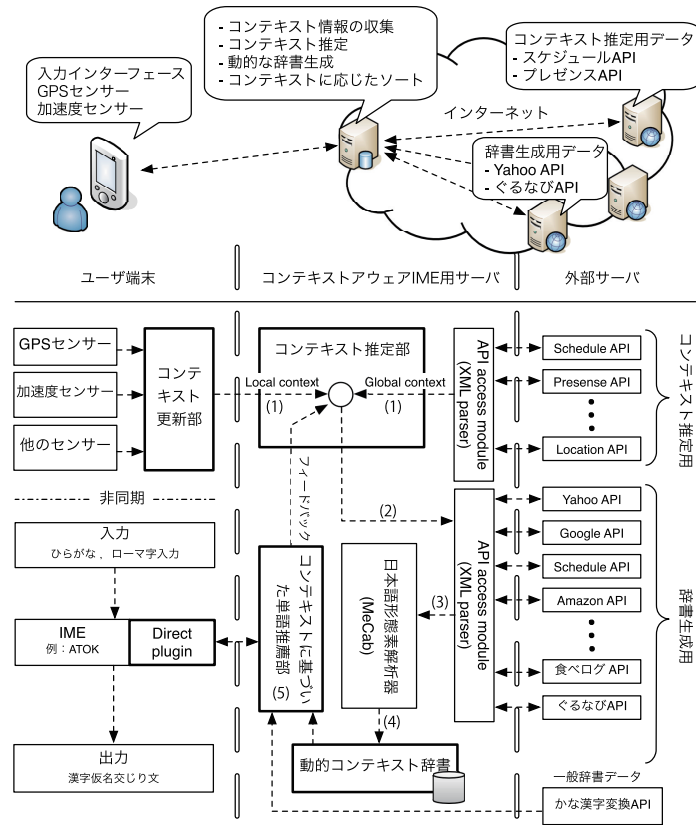


図 4 提案システムの構成
Fig. 4 Construction of our proposed system.

4.1 システムアーキテクチャ

図 4 に提案システムの構成を示す。提案システムは、ユーザー端末、インターネット上に設置されたコンテキストウェア IME 用サーバ、同じくインターネット上に公開されている Web サービス用のサーバ（本システム特有のものではない）から構成される。ユーザー端末は、GPS センサや加速度センサなど各種センサを備えるものとする。コンテキストウェア IME 用サーバは、(1) ユーザー端末上から得られる時刻や加速度センサデータなどのロー

カルコンテキスト情報と、ネットワーク上から得られるプレゼンス情報やスケジュール情報などのグローバルコンテキスト情報をもとにして、ユーザのコンテキストを推定する。次に、(2) 推定したコンテキストに応じて Web API へリクエストを送ることで単語を収集し、(3) 形態素解析器を通して、(4) 動的なパーソナルコンテキスト辞書を生成・更新する。最後に、動的に生成された辞書および通常の辞書の中から、(5) コンテキストを加味して、単語を選出し、Web 検索のヒット数に応じて並べ替えて、予測変換候補として表示する。本サーバの機能は、究極的にはローカルデバイス上に構築することも可能である。しかしながら、本提案システムでは、変換アルゴリズムの精度だけでなく、処理の速度も重要となることから、現在は、グローバルなネットワーク上にサーバとして設置する形態をとっている。さらに、より処理速度を改善するために、センサ情報に基づいた辞書生成と文字入力は非同期で行うようにし、位置が変化するたびにセンサ情報をアップロードし、Web API にアクセスして、辞書を更新している。その結果、文字入力時にはサーバ内の辞書データベース内の検索だけ済むため、辞書作成に用いる外部 Web サーバが増加したとしても速度の低下を防ぐことが可能なアーキテクチャとなっている。図 4 における外部サーバは、さまざまな企業が公開しているサーバであり、本システム特有のものではない。現在の実装システムでは、Yahoo!ローカルサーチ API²²⁾、Google Maps API²³⁾、ぐるなび API²⁴⁾ などと連携している。推定されたコンテキストをもとに、各 Web API へアクセスして漢字仮名交じりの文字列を取得し、形態素解析器 MeCab を用いた単語への切り分けと読み仮名の付与を経て、辞書の生成・更新が行われる。

4.2 ユーザのコンテキスト推定と学習

提案システムでは、さまざまな情報をもとに、ユーザが今どのような状態であるかというコンテキストを推定し、推定に基づいて適切と思われる文字列を変換候補として示す。コンテキスト推定手法に関しては、これまでさかんに行われている既存研究の成果を用いることを想定し、本論文では、代表的なコンテキストである位置情報と時間のみを取り扱う。位置情報を用いることにより、駅の近くにいる場合は最寄り駅名、現住所から遠く離れた場所にいれば近くのランドマーク名、書店や家電量販店にいれば商品名、というようにコンテキストに応じて単語の重要度が変化することが想像できる。ある位置で最寄り駅名を提示し、それをユーザが選択した場合、次回も同じ位置において最寄り駅名を選択する可能性が高いと考え、その位置に対する駅名の重みを上げる。また、選択しない場合は、位置に対して、駅名の重要度が低いことになる。このような学習を繰り返すことにより、ユーザにとって、その位置との相関が高い単語だけが、その位置で優先的に表示され、他の場所では一般的な

変換結果が示されるという理想的な状態にできると考えている。また、時間情報を用いることにより、現在放送中のテレビ番組のタイトルや出演者情報をもとにした単語を推薦したり、「帰る」や「おはよう」といった時間に依存すると考えられる単語を推薦することが可能となる。さらに、スケジュール API などと連携することで、予定に書かれたキーワードも候補として提示することが可能になる。

4.3 動的なパーソナルコンテキスト辞書の生成

提案システムでは、コンテキストに応じて、動的に辞書の作成を行う。アプローチとして、服部ら⁶⁾のように辞書に登録済みの単語に対してコンテキストとの相関を設定することにより、コンテキストによって並び順を変更するという事も考えられる。しかしながら、すべての単語に対して、コンテキストとの相関を設定するのは非常に困難である。そこで本研究では、標準辞書に加え、コンテキストに応じてそのつど関連のある Web API にアクセスし、単語を取得することにより動的に辞書を生成し、Social IME³⁾と同様にネットワーク上に配置する。現在では、Web サービスとして、さまざまな情報が公開され、Web API 経由で容易に取得することが可能である。今回利用する代表的な Web API としては、Yahoo!ローカルサーチ API²²⁾、Product Advertising API (Amazon 提供)²⁵⁾、ぐるなび API²⁴⁾、Google Maps API²³⁾ などがある。提案システムでは、これらから取得した漢字仮名交じり文を Mecab²⁶⁾により形態素解析し、単語ごとに分け、読み仮名を付けて、データベースに保存する。

4.4 予測変換候補のソート手法

単語のソートについては、各単語に与えられる初期スコア、および学習アルゴリズムの作成が必要である。提案システムでは、初期スコアとして、Yahoo!ウェブ検索 API を利用した検索結果数を用いる。Yahoo!ウェブ検索 API を用いることにより、その場所特有のニッチな単語よりも、その単語と同一の読みから始まる全国的に知名度の高い単語（たとえば、コンビニの名前）が上位に表示される可能性がある。しかしながら、今回は、待ち合わせメールやツイッターでの利用を想定し、“有名なランドマーク名ほどよく入力される”という仮定に基づき Yahoo!ウェブ検索 API を用いる。今後、大規模実験や Twitter のログ分析の結果に基づき、実際にその場所でよく利用される単語ほど上位に表示されるソートアルゴリズムに発展させる予定である。学習アルゴリズムは、単語の使用履歴として、従来時間に加えて、使用時のコンテキスト（本論文では位置情報）を記録しておき、次回同様のコンテキストが現れた際に同様の単語を推薦する。このように、推定と決定の繰返しによるフィードバックを取り入れることで、個人の行動パターンに基づいた予測変換の実現を目指

す。コンテキストの一致性に関しては、厳密な一致ではなく、ある範囲を設定する。たとえば位置情報の場合は、使用履歴に記録された位置 X に対して、現在位置 Y が X を中心にして距離 Z 内にいる場合は推薦する。

5. プロトタイプシステム

本章では、ネットワークと連携した日本語入力支援システムのプロトタイプに関して述べる。図 4 に示すように、提案システムは、ユーザ端末、コンテキストウェア IME 用サーバ、外部サーバから構成される。

5.1 ユーザ端末

今回作成したプロトタイプでは、ユーザ端末として、パソコンと携帯端末の 2 種類を作成した。また、位置情報に関しては、携帯端末に内蔵された GPS センサからの情報を用いることも可能であるが、今回は、デモンストレーションを行うために、Google Maps API を用いて、地図上で視覚的に位置を指定できるようにした。プロトタイプを実行する PC 上の入出力インタフェースとしては、デモ用として HTML フォームを用いたものと、既存の IME の拡張機能を用いたものの 2 通りを作成した。前者は、位置を変化させながら、動的辞書の候補となる単語の変化と、入力（ひらがな、またはローマ字）に対する出力（予測変換候補）の変化を見ることができる。後者は、ATOK ダイレクトプラグイン²⁷⁾として実装しており、PC 上のすべてのアプリケーションで、提案システムを利用することが可能になる。また、ATOK ダイレクトは、ATOK の Windows 版、Mac 版の両方で使える機能であるため、提案システムを両 OS で利用することが可能となっている。

図 5 に既存の IME である ATOK に実装したプロトタイプを示す。図 7 で指定した位置におけるランドマーク情報を ATOK のインタフェースで表示、入力できる。ATOK の仕様上、内部の一般的な辞書と本システムを同時に利用することはできないが、ファンクションキーを割り当てることにより、その切替えは十分な速度で行うことができる。

図 6 に、Android 上の OpenWnn を拡張したプロトタイプを示す。Android 端末としては、HT-03A (Android SDK 1.6) と HTC Hero (Android SDK 2.0) を用いている。OpenWnn を拡張することで、通常の変換候補とコンテキストウェア IME による変換候補を融合して表示することが可能となっている。この例では、分かりやすいように、コンテキストウェア IME による変換候補の後ろに通常の変換候補を表示しているが、それらを混ぜたり表示順序を制御したりすることも可能である。図中の位置は浜松駅を示しており、「は」という入力に対して、「浜松」という位置に基づいた単語が最上位に示されている。ま

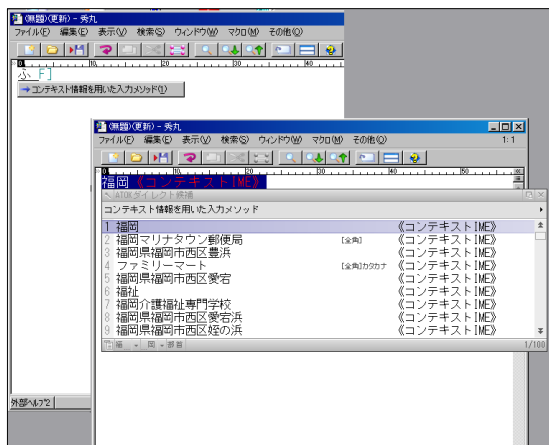


図 5 プロトタイプの実行画面 (ATOK ダイレクトプラグイン版)
Fig. 5 Screenshot of our prototype working as ATOK Direct Plugin.



図 7 コンテキストウェア IME 用サーバの状態表示
Fig. 7 Screenshot of status of the server for context-aware IME.



図 6 Android の OpenWnn 上への実装画面 (位置は浜松駅)
Fig. 6 Screenshot of our extended OpenWnn on Android.

た、「浜松イーストセブン」など漢字カタカナ交じりの長い単語も候補として表示することも可能となっていることが分かる。

5.2 コンテキストウェア IME 用サーバ

コンテキストウェア IME 用サーバでは、各種センサおよび Web API からの情報収集、

収集したデータからのコンテキストの推定、推定に基づく動的辞書の構築、コンテキストに応じた変換候補の推薦といった一連の処理を行う。本サーバの機能は、究極的にはユーザ端末上に構築することも可能である。しかしながら、本システムでは、変換アルゴリズムの精度だけでなく、処理の速度も重要となることから、現在は、グローバルなネットワーク上にサーバとして設置する形態をとっている。さらに、より処理速度を改善するために、センサ情報に基づいた辞書生成と文字入力とは非同期で行うようにしており、ユーザの位置情報の更新と辞書の生成を定期的に行っている。今回、実装言語は、ruby1.8 系を用い、サーバは Ubuntu9.06、データベースとしては SQLite を用いている。また、形態素解析器は、MeCab²⁶⁾を用いている。また、辞書生成用の Web API としては、Yahoo!ローカルサーチ API を使用した。図 7 は、コンテキストウェア IME 用サーバの状態を表示したものである。左側がユーザ端末の現在位置を指定する Google Map、中段に指定した位置に基づいて Yahoo!ローカルサーチ API から得られたランドマーク情報、右側にその文字列に対して形態素解析器で読み仮名をふり、Yahoo!検索 API のヒット数でソートした辞書の中身が表示されている。

6. 評価

本章では、提案システムの有効性を示すために行った評価について述べる。まず、マッシュアップシステムで問題となる応答速度に関して検証し、次に Web 検索ヒット数を用いたソート手法について検証する。最後に、間接的ではあるが、位置情報を含むインターネット上の文字列を分析し、位置情報と入力文字列に相関があることを示し、提案システムの有効性を示す。

6.1 応答速度に関して

提案システムでは、公開された外部 API を用いて、マッシュアップにより、動的に辞書を構築する。さらにその辞書はネットワーク上に配置される。ローカルにインストールされている IME と比較して、辞書生成時間と入力候補ダウンロード時間が付加され、応答時間が増大すると考えられる。携帯端末上での文字入力速度は、マルチタップ方式を用いた場合で 1 sec/letter 程度²⁸⁾ であり、これ以下の応答時間で動作することが望ましい。本システムでは辞書生成の際に形態素解析を行う必要があるが、モバイル端末上での使用を想定しているため、単体かつ軽量に動作することが望ましい。まず、辞書生成時間は、大きく分けて、1) 外部 API から文字列取得にかかる時間、2) 形態素解析により読み仮名をふる時間、3) Yahoo!検索 API によるランク付けを行う時間、から構成される。なかでも、2) にかかる時間が大きいことから、提案システムではコンテキストウェア IME 用のサーバを設置し、そのサーバ上に形態素解析器 MeCab²⁶⁾ を導入することで速度改善を行っている。さらに、辞書の更新タイミングを、ユーザの文字入力のタイミングと非同期にすることで、辞書生成時間をユーザに感じさせないようにしている。

今回の評価では、クライアントにおける「文字の入力」から「予測変換候補出力」までの間の応答時間 T を、外部 API を用いて辞書生成に関わる時間 T_d と、利用者の入力文字に基づいて辞書を引くために必要な時間 T_r と、得られた変換候補をユーザ端末に転送する時間 T_t に分けて分析を行った。 T_d に関しては、日本全国からランダムに選択した 266 カ所における結果を分析している。また、 T_r と T_t に関しては、そのうち 23 カ所において、それぞれ「あ (a)」「い (i)」「う (u)」「え (e)」「お (o)」で始まる単語を辞書から引いた結果を分析している。 T_t に関して、より現実の環境に近づけるため、通信路としては、E-mobile を用いている。

まず、 T_d に関する結果を表 1 と図 8 に示す。辞書生成時間は、Yahoo!ランドマーク API を用いたランドマーク情報取得と MeCab による形態素解析と読み仮名付与、ならびにデー

表 1 辞書サイズと辞書生成時間の結果

Table 1 Result of the size of generated dictionary and the time for generation T_d .

生成された辞書内単語数 (単語)			生成時間 (秒)		
最小	最大	平均	最小	最大	平均
38	503	238.8721805	0.487469	3.019088	1.860482139

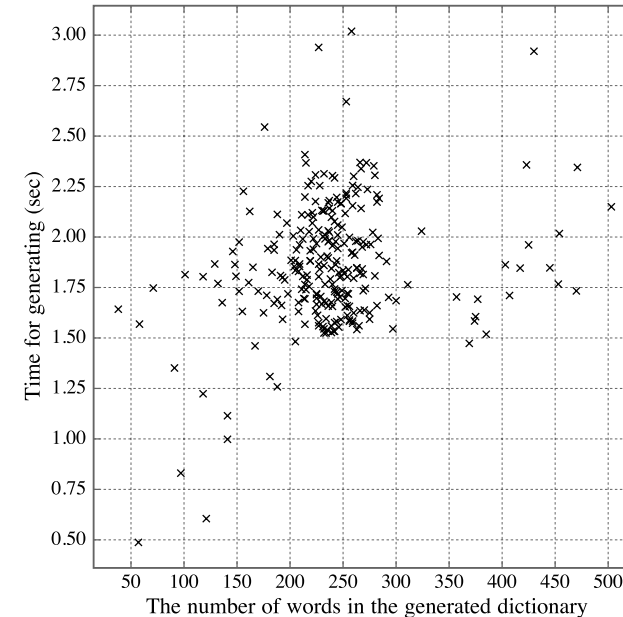


図 8 辞書サイズと生成時間の分布

Fig. 8 Distribution of dictionary generation time for the size.

タベース書き込み時間から構成される。表 1 より、生成された辞書内の単語数は、最小で 38 単語、最大で 503 単語、平均は 238.9 単語であることが分かる。また、その生成時間は、最小で 0.5 秒、最大で 3 秒程度、平均で 1.9 秒ほど要していることが分かる。Yahoo!ランドマーク API で取得可能なランドマーク名は、1 座標につき最大 100 件であるが、これを MeCab により、複数の単語に分割しているため、辞書サイズが大きくなっている。MeCab の辞書としては、NAIST JDIC (0.6.2-20100208) を用いる。たとえば、「九州大学伊都キャンパス」は、「九州大学伊都キャンパス」だけでなく、その構成要素である「九州大学」「伊

表 2 候補単語数に対する T , T_r , T_t の結果

Table 2 Result of T , T_r and T_t for the number of candidate words.

	最小	最大	平均
提示された候補数 (単語)	0	24	6.026785714
総所要時間 T (秒)	0.469193 (1 単語)	15.26841 (18 単語)	0.848508813
辞書読み取り時間 T_r (秒)	0.186305 (19 単語)	0.264816 (18 単語)	0.192301478
データ転送時間 T_t (秒)	0.277089 (5 単語)	15.056005 (5 単語)	0.656213214

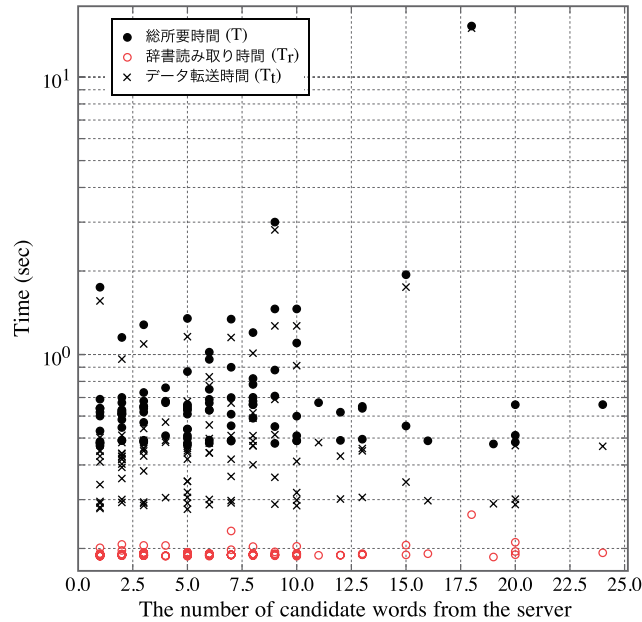


図 9 候補単語数に対する T , T_r , T_t の分布

Fig. 9 Distribution of T , T_r and T_t for the number of candidate words.

都」「キャンパス」が辞書に登録される。これらの結果から、辞書の生成には、数秒の時間がかかるため、ユーザからの文字入力時に辞書を生成することは難しいことが分かる。我々は、辞書生成を実際の文字入力とは非同期に行うことで、 T_d をユーザに感じさせないようにしている。

一方、図 8 は測定した 266 カ所に関して、生成された辞書サイズに対する生成時間の分

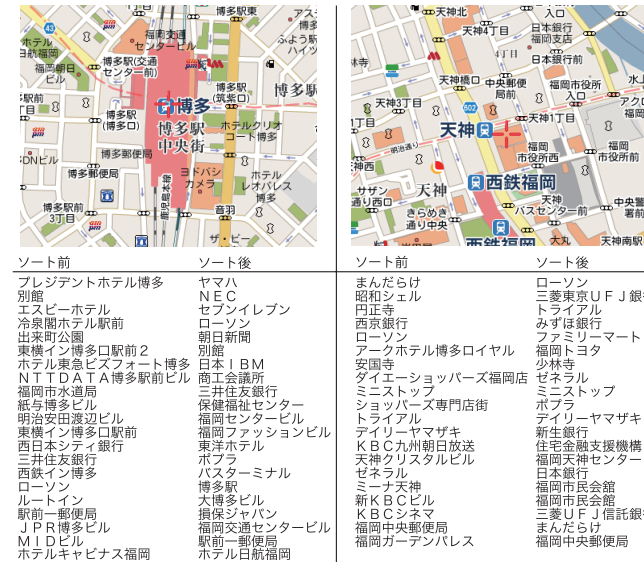


図 10 評価シートの一例

Fig. 10 Examples of evaluation sheet.

布を示す。この結果から、辞書が小さい場合には多少生成速度が速いが、明確な相関はみられないことが分かる。

次に、表 2 と図 9 に生成された辞書を引くのに要する時間を示す。表 2 から、ユーザに提示される単語は、最小で 0 語、最大で 24 語、平均で 6 語程度であることが分かる。これは、今回、あ行で始まる単語のみを評価しているため、場所によっては、あ行で始まる単語が 1 つも登録されていない可能性があるためである。ユーザが先頭文字を入力して、予測候補として表示されるまでの時間は、最小で 0.5 秒、最大で 15 秒かかっていることが分かる。この中で、辞書読み取り時間 T_r は、最小で 0.19 秒、最大で 0.26 秒と分散が小さく、データ転送時間 T_t が支配的であることが分かる。今回、E-mobile を用いて実験を行ったが、実験中の電波環境の変化により、このようなバラツキが出たと考えられる。

次に、図 9 に候補単語数に対する T , T_r , T_t の分布を示す。横軸は辞書のサイズを示している。まず、 T_r に着目すると、辞書サイズが大きくなってもその辞書を引く時間はほとんど変化がないことが分かる。このことから、 T は通信路の遅延 T_t に大きく依存してい

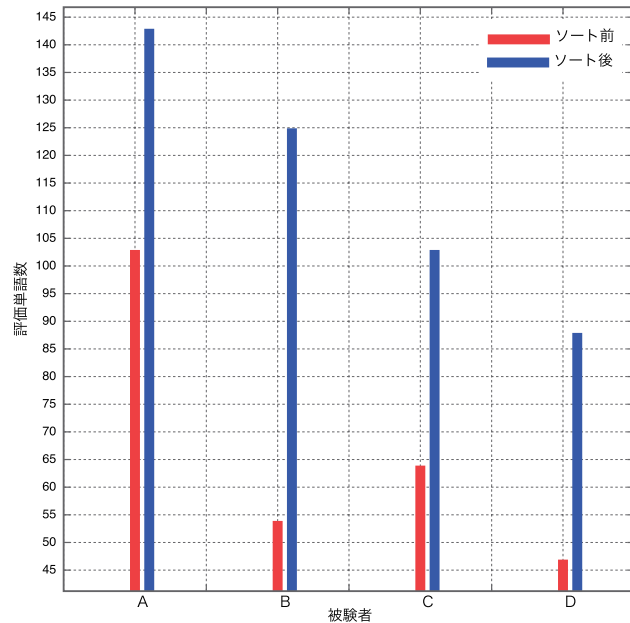


図 11 ユーザ別の有用単語数

Fig. 11 The number of useful words by users.

ることが分かる。

6.2 ソート手法に関して

提案システムで用いる Web 検索ヒット数に基づいたソート手法の有用性に関して、アンケートと、インターネット上の文字列の分析から評価する。後者に関しては、次節において、位置と入力文字列の相関関係とともに説明する。アンケートでは、予測変換候補上位 20 件について、ソートを行う前と後の結果をユーザに提示し、入力する可能性のある単語の個数を数え上げてもらった。この単語の個数をソートの前と後のリストで比較し、ソートされた予測変換単語の有用度の検証を行った。図 10 にユーザへ提示した評価シートの一部を示す。評価シートには予測変換候補として表示される単語リストと、ユーザの現在地を示す地図が表記されている。単語リストは 2 種類あり、それぞれ地図上のマーカ周辺のランドマーク情報を API の出力のまま表示したもの（ソート前）と、それに対して Yahoo!ウェブ検索 API によるスコア付を行ってソートしたものとなっている。ユーザはそれぞれ 20 個の

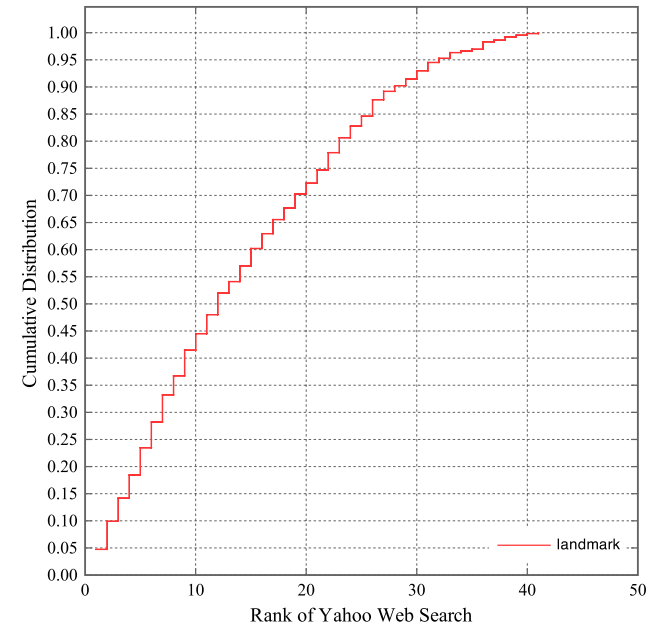


図 12 一致した文字列の Yahoo!検索を用いた順位付けの累積密度分布

Fig. 12 Cumulative distribution of the rank of corresponding words sorted by Yahoo! Search.

単語から構成される単語リストから、有用と思われる単語の個数を数える。1 つの場所を 1 セットとし、合計 10 セット（計 400 単語）用意してある。4 人のユーザに対してアンケート調査を行い、最終的に得られたリストごとの値を評価値として使用した。図 11 に示すように、検索数によるソートを行った場合、単語数にして約 1.7 倍程度有効な単語を推薦できることが分かった。

また、別の観点から、Yahoo!検索 API に基づいたソート手法を評価する手法として、予備実験で収集した Twitter のログを用いた分析を行った。まず、ツイートごとに、つぶやかれた位置において提案システムが推薦されるであろう文字列群とマッチングを行う。もし、推薦文字列群の 1 つがツイートに含まれている場合は、推薦文字列群に含まれるすべての文字列に対して、Yahoo!検索 API によるスコアを算出し、含まれていた文字列の推薦順位を分析する。図 12 に、一致した文字列の推薦順位の累積密度分布を示す。この図から、一致した文字列が提案ソート手法において上位 10 位以内に含まれる確率が約 45%であり、得

られた文字列の中から, web 検索結果の総数に基づき絞り込む手法は有用性が高いと考えられる.

7. おわりに

本論文では, 携帯端末における新たな省入力化のアプローチとして, コンテキストに基づいて動的に辞書を生成するコンテキストウェア IME を提案し, PC および Android 端末上に実装した. まず, 予備実験として, Twitter 上の位置情報付きツイートを分析した結果, 入力文字列と位置に関係性があることを確認した. 実装システムでは, Google Map 上で指定したユーザの位置に応じて, Yahoo!ローカルサーチ API を用いて動的に辞書を作成する. 実装システムは, 辞書生成の非同期化と MeCab を導入したサーバの設置により, キー入力から候補提示まで平均約 0.85sec という応答速度を達成し, 十分実用に耐えうる速度で動作することを示した. また, Web 検索ヒット数を用いたソート手法を提案し, 被験者によるアンケートから, 提案手法により有用度の高い単語を推薦できることを明らかにした. 今後の課題として, 被験者を用いた大規模実験により, 動的に生成された辞書の利便性に関して実際に検証する予定である. そして, 最終的には, 実際にその場所によく利用される単語を上位に表示できるシステムを目指す.

参 考 文 献

- 1) rTYPE: 「ネットは PC より携帯」携帯ネット歴 5 年以上では半数以上—rTYPE アイシェアオンラインリサーチサービス市場調査公開 (2009).
<http://release.center.jp/2008/11/0502.html>
- 2) Grover, D., King, M. and Kuschler, C.: Patent No.US5818437, Reduced keyboard disambiguating computer, Tegic Communications, Inc., Seattle, WA (1998).
- 3) 奥野 陽, 萩原将文: インターネットを用いた日本語入力システム, 情報処理学会第 190 回自然言語処理研究会 (2009).
- 4) Masui, T.: POBox: An Efficient Text Input Method for Handheld and Ubiquitous Computers, Lecture Notes in Computer Science, Vol.1707/1999, pp.289–300, Springer Berlin / Heidelberg (1999).
- 5) オムロンソフトウェア株式会社: iWnn. <http://www.omronsoft.co.jp/SP/>
- 6) 服部 峻, 手塚太郎, 田島敬史, 田中克己: モバイル・ユビキタス環境のための実世界コンテキストに基づく検索質問修正, 電子情報通信学会第 17 回データ工学ワークショップ (DEWS2006), pp.2B–i10 (2006).
- 7) 中村一正: Web 上での文章入力時における情報補完, 修士論文, 北陸先端科学技術大学院大学 (2007).

- 8) 小松弘幸, 高林 哲, 増井俊之: 動的略語展開を利用した文脈をとらえた予測入力, 情報処理学会論文誌, Vol.44, No.11, pp.2538–2546 (2003).
- 9) 小松弘幸, 高林 哲, 増井俊之: 文書蓄積システム Kukura を用いた予測入力, 日本ソフトウェア科学会 WISS2002, pp.43–47 (2002).
- 10) 猿田芳郎, 富井尚志: 加速度センサと RFID を用いたユビキタス環境での利用者コンテキスト推定手法, 日本データベース学会 (DBSJ) 論文誌, Vol.6, No.3, pp.13–16 (2007).
- 11) 森岡英之, 岩田直樹, 新津善弘: 複数センサを用いたコンテキスト推定法, 電子情報通信学会講演集, 講演番号 94 (2008).
- 12) 林 智天, 川原圭博, 田村 大, 森川博之, 青山友紀: 小型モバイルセンサを用いたコンテキスト適応型コンテンツ配信サービスの設計と実装, 電子情報通信学会技術研究報告 IN, Vol.104, No.691, pp.149–154 (2005).
- 13) 麦嶋慎也, 清水隆司, 富井尚志: ユビキタス環境 DB における利用者の概念を利用した行動支援手法, 日本データベース学会 (DBSJ) 論文誌, Vol.7, No.1, pp.221–226 (2008).
- 14) 天野 翔, 川原圭博, 森川博之, 青山友紀: 実空間情報を統一的に取り扱うミドルウェアの設計, 電子情報通信学会総合大会講演論文集通信 (2), Vol.2006, No.2, p.264 (2006).
- 15) 森川博之: 実空間指向ユビキタスネットワーク, 電子情報通信学会和文論文誌 B, Vol.88, No.11, pp.2137–2146 (2005).
- 16) 川原圭博, 司 化, 森川博之, 青山友紀: コンテキストウェアサービスプラットフォーム Synapse の設計と実装, 電子情報通信学会総合大会講演論文集通信 (2), Vol.2005, No.2, p.537 (2005).
- 17) 松方正樹, 星川恭子, 瀧 浩平: コンテキストウェアネスとその応用, 沖テクニカルレビュー, Vol.75, No.1, pp.94–97 (2008).
- 18) 宮崎雄一朗, 山田直治, 住谷哲夫: ユーザの行動に合わせたサービス実現のための行動推定技術の開発, NTT DoCoMo テクニカル・ジャーナル, Vol.17, No.3, pp.55–61 (2009-10).
- 19) Nishio, N., Sakamoto, N. and Arai, I.: Real-World-oriented Contents Management System Mashed up with Google Street View, *Demonstration at Pervasive* (2009).
- 20) Twitter 社: Twitter. <http://twitter.com/>
- 21) 荒川 豊, 田頭茂明, 福田 晃: Twitter におけるコンテキストと単語の相関関係分析, 情報処理学会研究報告, Vol.2010-MBL-53, No.50, pp.1–7 (2010).
- 22) ヤフー株式会社: Yahoo!デベロッパネットワーク — 地図 (2009).
<http://developer.yahoo.co.jp/webapi/map/>
- 23) グーグル株式会社: Google Code (2009). <http://code.google.com/intl/ja/apis/maps/>
- 24) 株式会社ぐるなび: ぐるなび Web サービス (2009). <http://api.gnavi.co.jp/api/>

service.htm

- 25) アマゾンジャパン株式会社: Product Advertising API (2009).
<https://affiliate.amazon.co.jp/gp/advertising/api/sdetail/main.html>
- 26) 工藤 拓: MeCab: Yet Another Part-of-Speech and Morphological Analyzer (2009). <http://mecab.sourceforge.net/>
- 27) Corporation, J.: ATOK.com — 日本語入力システム「ATOK (エイトック)」や日本語に関する情報のサイト (2009). <http://www.atok.com/>
- 28) 安藤明伸, 川野常夫, 田村 博, 長谷川聡, 宮尾 克: モバイル・ヒューマンインタフェースの動向, ヒューマンインタフェース学会誌, Vol.8, pp.23-32 (2006).

(平成 22 年 5 月 21 日受付)

(平成 22 年 10 月 4 日採録)



荒川 豊 (正会員)

1977 年生。2001 年慶應義塾大学理工学部情報工学科卒業。2003 年同大学大学院修士課程修了。2004 年同大学 COE 研究員。2006 年同大学大学院博士課程修了。博士 (工学)。2006 年同大学院特別研究助手。2007 年同大学院特別研究助手。2009 年 3 月より九州大学大学院システム情報科学研究科助教。2010 年 4 月より同大学システム LSI 研究センター助教 (兼務)。主として、ネットワークプロトコル, トラフィック制御, 次世代ネットワークアプリケーションに関する研究に従事。APCC 2008 Best Paper Award (2008 年), 情報処理学会 MBL 研究会優秀論文賞 (2009 年), DICO MO 優秀論文賞 (2010 年), DICO MO 優秀プレゼンテーション賞 (2010 年)。IEEE, 電子情報通信学会各会員。



末松 慎司

1985 年生。2008 年九州大学工学部地球環境工学科卒業。2010 年同大学大学院システム情報科学府修了。コンテキストウェアシステム, 日本語入力システムに関する研究開発に従事。同年エヌ・ティ・ティ・コミュニケーションズ株式会社に入社。



田頭 茂明 (正会員)

1996 年龍谷大学理工学部卒業。1998 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。2000 年同大学情報科学研究科博士後期課程修了。博士 (工学)。2000 年広島大学工学部助手。2007 年同大学大学院工学研究科助教。同年九州大学高等研究院特別准教授, 同大学大学院システム情報科学研究科特任准教授。モバイル・ユビキタスコンピューティング, システムソフトウェアの研究に従事。情報処理学会山下記念研究賞 (2009 年), 電子情報通信学会通信ソサイエティ活動功労賞受賞 (2009 年)。IEEE, 電子情報通信学会各会員。



福田 晃 (フェロー)

1977 年九州大学工学部情報工学科卒業。1979 年同大学大学院工学研究科修士課程情報工学専攻修了。同年日本電信電話公社 (現 NTT) 武蔵野電気通信研究所入所。1983 年九州大学助手。1989 年同大学助教授。1994 年奈良先端科学技術大学院大学教授。2001 年九州大学大学院システム情報科学研究科教授, 2008 年九州大学システム LSI 研究センター長 (兼任), 現在に至る。工学博士。組み込みソフトウェア, ユビキタスコンピューティングに関する研究に従事。情報処理学会研究賞 (1990 年), Best Author 賞 (1993 年) 等を受賞。情報処理学会フェロー, 電子情報通信学会, ACM, IEEE Computer Society, 日本 OR 学会各会員。