

ユーザ生成情報を用いた携帯端末上での 状況依存型サービス推薦

矢野 幹樹^{†1} 白木 敦夫^{†1} 梶 克彦^{†1}
松原 茂樹^{†1} 河口 信夫^{†1}

近年、モバイル端末向けに提供される Web サービスやアプリケーションなどのサービスの数は爆発的に増加しており、膨大な数のサービスの中から自分に必要な物を探し出すことが困難となりつつある。そこで我々は、ユーザの状況に応じてサービスの推薦を行うシステムの提案を行う。提案するシステムは、自分の現在状況をクエリとする検索サービスを提供する段階と、そこから収集した学習情報を用いて自動的な状況推定を行う段階の 2 つの段階に分けてサービス提供することで、ユーザから学習情報を収集しつつ状況に応じたサービスの推薦を実現していく事ができる。本論文では、提案するシステムの第 1 段階のプロトタイプを実装し、評価実験を行った。その結果、提案手法によりユーザが入力した状況に適したアプリを推薦できることを確認した。

Context-Aware Mobile Service Recommendation System using Consumer-generated Information

MOTOKI YANO,^{†1} ATSUO SHIRAKI,^{†1}
KATSUHIKO KAJI,^{†1} SHIGEKI MATSUBARA^{†1}
and NOBUO KAWAGUCHI^{†1}

Recently, it becomes very hard for users to find their desired mobile services because the number of applications and Web services are rapidly increasing. Therefore, we propose a system for recommending services according to the user's context. The development of this system is divided into two stages. At the 1st stage, we provide a searching service for mobile applications using user's context as query. 2nd stage, system automatically estimates user's context based on the collected query logs from the 1st stage. In this paper, we have implemented a prototype system of the 1st stage and evaluated the effectiveness of this system.

1. はじめに

近年、携帯電話やスマートフォンといった高性能な携帯端末が広く普及し、いつでも、どこでも多様なサービスを利用する事ができる環境が整いつつある。そしてこれに伴い、モバイル端末向けに提供される Web サービスやアプリケーション（アプリ）などのサービスの数は爆発的に増加している。しかし、サービスの数が増加するほどに、膨大な数のサービスの中から自分に必要な物を探し出すことは困難となる。

このような問題に対し、ユーザの状況に応じてサービスを推薦することで、ユーザにとって必要なサービスの提示を行うことが可能であると考えられる。そこで本研究では、サービスの中で特にモバイル端末向けのアプリを対象として、アプリをユーザの状況に応じて推薦する手法を提案し、システムを実現することを目的とする。想定する例として、

- 駅の近くでの時刻表アプリの推薦
- 繁華街の近くでの飲食店情報を検索するアプリの推薦

などが挙げられる。

ユーザの状況に応じたサービスの推薦に関するこれまでの研究では、コンテンツの特徴と、ユーザのコンテキスト情報（位置や時間など）との距離に基づく推薦¹⁾ や、類似した状況にいたユーザが利用したコンテンツを推薦する協調フィルタリングによる推薦²⁾ などの手法が提案されている。しかし、これらの手法はあらかじめ大量の学習データが必要となり、学習データが無ければ推薦を行うことができない。

十分な学習データを収集するためには大きな手間を要するが、有用なサービスを多くのユーザに利用してもらうことを通じて暗黙的に学習情報を取得することができれば、大量の学習情報をユーザの負担無しに収集することができると考えられる。そこで、本研究では、推薦サービスを 2 段階に分けて段階的に発展させながら、ユーザから提供されるアプリの利用状況情報に基づいて、状況に応じたアプリの推薦を実現することを目指す。まず、第 1 段階では、ユーザに自分の現在状況をキーワードとして表現した「状況タグ」をクエリとするアプリ検索サービスを提供する。そして、状況タグとコンテキスト情報との対応を記録する。次に、第 2 段階では、第 1 段階にて取得した状況タグとコンテキスト情報との関係を利用することにより、コンテキスト情報から状況タグを機械的に推定し、アプリの推薦を行う。

^{†1} 名古屋大学
Nagoya University

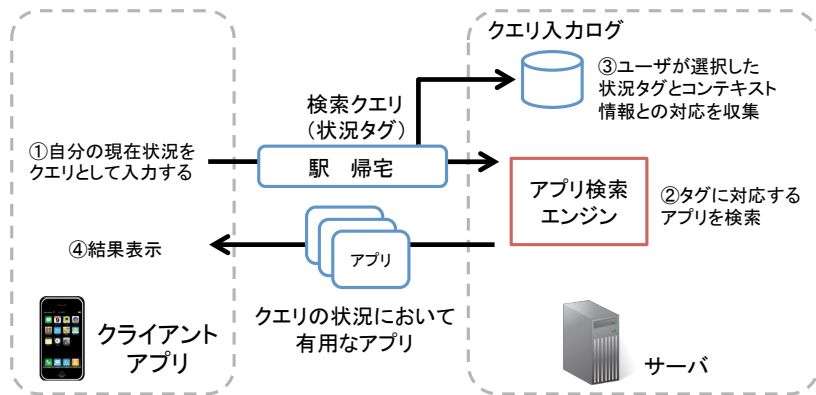


図 1 第 1 段階：自分の現在状況をクエリとするアプリ検索サービス

Fig. 1 1st stage: Searching service for applications using user's current context as query

このような段階的な手法により推薦を行うことで、状況を推定するために必要な情報が無い状態でもユーザーに有用なサービスを提供することができ、そのサービスを多くのユーザーに利用してもらうことを通じて大量の学習データを収集することができる。そして、最終的には自動的に状況を推定しサービスを推薦するシステムを実現することができると考えられる。

以下、2 節では提案するサービス推薦システムの概要について述べ、3 節、4 節にて、サービス推薦の各段階についての詳細を述べる。5 節では提案するシステムの第一段階に関する実験と評価を行い、6 節でまとめを行う。

2. 推薦システムの概要

提案するシステムは 2 つの段階に分けられる。第 1 段階の概要を図 1 に、第 2 段階の概要を図 2 に示す。両段階とも、ユーザーが利用する端末上で動作するクライアントアプリと、アプリの検索及びコンテキスト推定を行うサーバにより構成される。

まず、第 1 段階では、ユーザーに対して、「自分の現在状況をクエリとするアプリ検索サービス」を提供する。これは、ネット上で情報検索を行うように、ユーザーが今自分がどのような状況にいるのかをキーワードとして入力してもらい、その状況にマッチするアプリを提示するというサービスである。そして、このサービスをユーザーに利用してもらうことを通じ

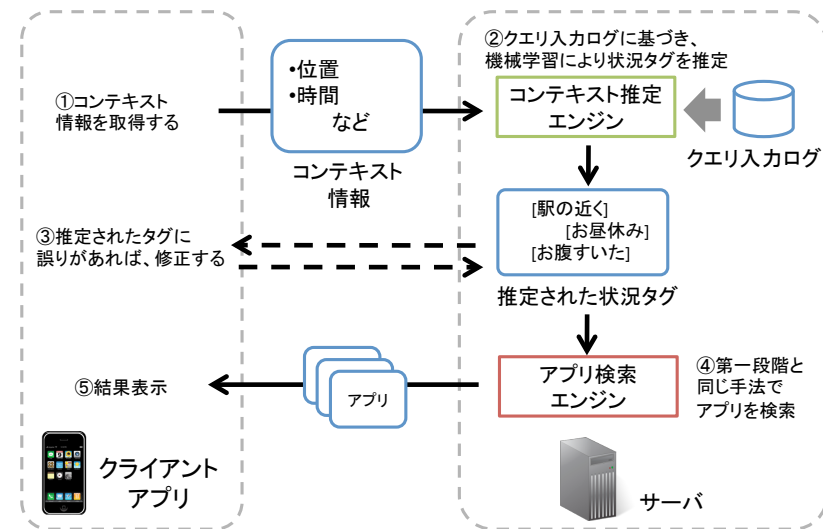


図 2 第 2 段階：クエリ入力ログに基づく状況依存型サービス推薦

Fig. 2 2nd stage: Context-aware service recommendation based on the collected query logs

て、ユーザーがクエリを発行した際に端末側でコンテキスト情報を取得し、クエリと同時にサーバ側へと送信してデータベースに蓄積する。「コンテキスト情報」とは、位置、時間など、端末のセンサなどからユーザーの入力操作無しにプログラムが取得可能なユーザーの状況に関する情報であるとする。

次に第 2 段階では、端末上で取得したコンテキスト情報から、その状況において入力され得る状況タグを推定し、アプリ検索サービスの検索画面に推定されたタグを自動的に入力する。そして、第 1 段階と同じ手法によりアプリの推薦を行う。この際、タグの推定は、第 1 段階で取得したクエリ入力ログを学習データとした機械学習アルゴリズムを用いて、過去に似た状況にいたユーザーが最も多く入力したキーワードを解析することで行う。ユーザーはこれにより、第 1 段階のように自分から状況タグを入力する必要無しに、その場において有用なアプリの推薦を受けることが可能になると期待できる。

このように、2 段階に分けて推薦サービスを実現することにより、状況の推定に必要な学習情報が十分に無い状態であっても、アプリ検索サービスとしてユーザーに有用なサービスを

提供することができると考えられる。そして、アプリ検索サービスによって十分な学習情報が収集された時点から、ユーザの状況を自動的に推定してアプリを推薦するサービスへとシームレスに移行していくことができると考えられる。また、第1段階にて入力されるクエリは、アプリの検索に特化したものではないため、収集されたクエリ入力ログはアプリ推薦に限らず、他のあらゆる状況依存型サービスの構築において利用することができるという利点がある。

以下、それぞれの段階の具体的な実現方法を述べる。

3. 第1段階：ユーザの現在状況をクエリとするアプリ検索サービス

本節では、ユーザの現在状況をクエリとするアプリ検索サービスを実現するにあたり、各アプリが有用となる状況を知る手法として、Web上に存在するユーザ生成情報を利用する方法を提案する。また、クエリの入力方法、及びアプリの検索とランキングの手法の提案を行う。そして、このサービスの運用を通じて、第2段階に必要な学習データとしてクエリ入力ログを収集することについて述べる。

3.1 ユーザ生成情報の利用

ユーザが入力する状況を表すタグからその状況に適したアプリを検索するためには、アプリがどのような状況で有用であるかという情報をあらかじめ収集しておく必要がある。iPhoneやAndroid向けアプリを提供するアプリケーションマーケットであるApp Store³⁾やAndroid Market⁴⁾などでは、アプリ作者はアプリを紹介するコメントと共にアプリを公開する事ができるが、どのような状況で有用であるのかという情報を取り分けて記述するような仕組みにはなっていない。アプリ一つ一つに対して、どのような状況で有用であるかというキーワードを、推薦サービス提供者やアプリ作者に付与してもらうなどといった方法も考えられるが、iPhoneでは既にアプリが20万個以上存在するという現状において、あまり実現可能な方法とは言えない。

そこで本研究では、Web上に存在する、ユーザによって生成されたアプリのレビューを利用する。App StoreやAndroid Marketには、アプリに対してユーザがそのアプリのレビューを投稿できる仕組みが一般的に備えられている。また、Web上には、ブログ等にアプリのレビューを掲載するユーザが多数存在する。これらWeb上に存在するアプリのレビューの内容としては、主にユーザがアプリを使った感想や、他人へのお勧めコメント、また、作者への要望などが挙げられる。このような文章中では、アプリがどのような状況で有用なのかという記述や、どのような人に対して有用なのかといった記述などが含まれている

場合がある。

例として、「駅.Locky」というiPhone向けアプリのレビューの一部（App Storeにて投稿されたもの）を以下に示す。駅.Lockyは、我々の研究室で開発・運用を行っている位置依存型時刻表アプリで、電車の発車までの時間をカウントダウン形式で表示するものである⁵⁾。

- 駅.Lockyは次の電車の出発までの時間をカウントダウン形式で表示する時刻表アプリです。時刻表データを端末内に保存する事ができるので、ネットに接続していなくても、目的の時刻表を素早く見る事ができます。
- 朝はいつも乗る電車の時間までのカウントダウンに、帰りは乗る電車がバラバラなので、学校を出るときに利用して駅で無駄な待ち時間がでないように使ってます。作者さんと、ダイヤデータを作ってくれた方に感謝です。

上記のレビュー中には、「駅」や「電車」、「朝」、「帰り」、「学校を出るとき」などの言葉が含まれている。このアプリは、発車までの時間をカウントダウンするという性質上、駅の近くや電車の中で目的の電車の発車までの時間を調べたり、通勤、通学時に自宅や勤務先等で普段利用する電車の発車時刻を調べたりするといった使い方が有用であると想定される。そのため、前述したレビュー中に含まれる言葉は、このアプリが有用である状況を表したものであると考える事ができる。

3.2 クエリの入力方法

ユーザに自分の現在状況を検索クエリとして入力してもらうにあたり、ユーザにとっての入力しやすさ、そして、システム側で検索を行い易い方法として、以下の2つの入力方法を挙げる。各入力手法のイメージを図3に示す。

3.2.1 フリーワード入力

一般的な情報検索と同じく、検索キーワードとして状況タグをユーザ自身で作成して入力する方法である。ユーザには自由に入力可能なテキストボックスを提示し、その中にスペースなどを区切りにして状況タグを1つないし複数入力してもらう。

この方法は、ユーザによって自由な状況タグを記述することができるため、多様に存在するユーザの状況を、言語の表現の範囲内で全て表現することができるというメリットが考えられる。しかし、表現が自由すぎると、同じ状況であってもユーザによって状況の表現の仕方が異なり、タグの種類が無限に発散してしまう問題が考えられる。また、文字入力がしにくいモバイル端末上で、文字入力することは手間が多い。そのため、ユーザに気軽に現在状況を入力して検索を行ってもらうことができないおそれがある。



フリーワード入力

タグ選択入力

図 3 クエリ入力方法のイメージ

Fig. 3 Images of query input method

3.2.2 タグの選択による入力

あらかじめ状況タグの候補を用意しておき、ユーザにその中から自分の状況を表現したものを選択してもらうという方法である。画面上には、ユーザが入力するであろうタグの候補を、タグクラウドと呼ばれる形式で表示する。そして、その中からタグを1つないし複数選択してもらう。

この方法は、文字入力の手間が無く、近年のスマートフォンで主流となりつつあるタッチインタフェースにおいて入力のしやすい方法であるといえる。また、あらかじめ用意されたタグの中だけから選択を行うため、ユーザによって表現にばらつきが生じる事が無いというメリットがある。しかし、状況の表現範囲は用意されたタグの種類に限定されるため、あらかじめ十分なタグの種類を用意しておかなければならないという問題がある。また、画面上に表示させる事ができるタグの数には限りがあるため、抽象的な状況を表すタグから、より具体的な状況を表すタグへと階層的に選択できる仕組みが必要であると考えられる。

我々は、状況を表すタグをあらかじめ用意しておくために、Twitterなどのマイクロブログに投稿されたつぶやきを解析し、状況を表す言葉（状況語）の辞書や、抽象的な状況語から具体的な状況語への階層関係を生成する手法の検討を本研究と並行して行なっている⁶⁾。

3.3 アプリの検索とランキング

レビューなどのユーザ生成情報から、検索クエリの示す状況に適したアプリを探し出し、かつ、その状況において関連性の高い順にランキングするための具体的な手法として、以下の2つの手法を挙げる。

3.3.1 ワードを含むレビュー数によるランキング

あるアプリに関する全レビュー中に、与えられた検索クエリ中のキーワードを含むレビューがどれだけ含まれるかでランキングを行う。具体的には、あるアプリ a の全レビュー数を $|R_a|$ 、そのうち与えられたキーワードのうちいずれかを含むレビュー数を $|\{r : r \in w_i\}|$ とした時、アプリ a の検索スコア S_a を、

$$S_a = \frac{|\{r : r \in w_i\}|}{|R_a|} \quad (1)$$

として定義する。そして、スコア S_a が大きい順にアプリを並べ替える。尚、アプリ a の全レビュー中に、与えられた検索ワードが1つでも含まれない場合、 a はランキングから除外する。

3.3.2 tf-idfの結果を用いたランキング

与えられた検索クエリ中のキーワードの、あるアプリに関する全レビューから算出したtf-idfの値に基づいてランキングを行う。単語 w_i のtf-idfは、以下の式によって求められる。

$$tfidf = tf \cdot idf \quad (2)$$

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (3)$$

$$idf_i = \log \frac{|D|}{|\{d : d \in w_i\}|} \quad (4)$$

n_i はアプリ a の全レビュー中の単語 w_i の出現頻度、 $|D|$ は全アプリの総レビュー数、 $|\{d : d \in w_i\}|$ は単語 w_i を含むレビュー数である。

3.4 アプリ検索サービスからの学習データの収集

提案する検索サービスでは、ユーザが入力した自分の現在状況を表すクエリをサーバに送信する際に、端末で取得可能なコンテキスト情報（位置や時間など）を匿名の情報として送信する。サーバ側は、これらのクエリとコンテキスト情報を受け取り、データベースに保存

する。これにより、状況を表すタグとコンテキスト情報との結びつきの情報を蓄積する事ができる。

4. 第2段階：クエリ入力ログに基づく状況依存型サービス推薦

第1段階で収集した情報には、クエリ入力ログとその状況タグが入力された際のコンテキスト情報との対応が記録されているため、この情報を利用することで、コンテキスト情報から、その情報に近い状況において他のユーザが入力した状況タグを推定する事ができると考えられる。そこで、第2段階では、収集された情報に基づいて、端末より取得した現在のコンテキスト情報からユーザの状況タグを推定し、その状況に適したアプリを自動的に推薦する。その際、推定された状況タグをユーザに一度提示し、推定結果の確認と修正を行ってもらうことで、フィードバックを獲得し推定精度の向上に用いる。

以下、現段階で想定している各操作に関する詳細を述べる。

4.1 コンテキスト情報からの状況タグの推定

ユーザが利用する端末において取得したコンテキスト情報は、サーバ側へと送信され、サーバ側では、受け取ったコンテキスト情報を、状況推定エンジンに入力し、出力結果として状況タグを得る。

状況推定エンジンは、クエリ入力ログを学習データとする機械学習アルゴリズムによって構築する。機械学習アルゴリズムに関しては、現段階ではまだ検討中である。どのような機械学習アルゴリズムを利用するかは、利用するコンテキスト情報の値の特徴や、実際の推定結果の評価に基づいて決定する必要があると考えられる。

4.2 推定された状況タグのユーザによる選別

状況推定エンジンにより推定された状況タグのリストは、一度クライアント側へ送信される。クライアント側では、3.2節で示したタグ選択インターフェース上に、推定されたタグのリストを提示する。これに対しユーザは、提示されたタグが自分の状況に当てはまるかどうかを確認し、当てはまらないタグを候補リストから除外してもらう。また、当てはまるタグが存在しない場合は、通常のタグ選択と同じように、自分の状況を表すタグをユーザ自身の手で選択してもらう。選別された状況タグは再度サーバへ送信される。

このように、推定されたタグの候補を一度ユーザに提示し選別を行ってもらうことで、推定結果に誤りがある場合でも、ユーザ自身の判断でタグを結果を修正することができ、ユーザの意に沿った推薦を行うことができる。また、ユーザが修正を行った情報から、推定結果に対するフィードバックを得ることができ、推定エンジンの精度を向上させることができる

と考えられる。

4.3 推定された状況タグからのサービス推薦

ユーザによるタグの選別後、再度送信されたタグの情報に基づいて、サーバ側でそのタグに対応するアプリの検索を行う。この検索は、3.3節で示したアプリの検索とランキングの手法と同じ手法で行う。ランキングの結果はクライアント側へと送信され、クライアントアプリ上でアプリのリストが提示される。

5. 実験

今回、第1段階であるユーザの現在状況をクエリとするアプリ検索サービスについて、提案した手法がどの程度有効であるかの評価を行った。具体的には、ユーザがある状況において自分の状況を検索クエリとして入力した際、その状況においてユーザが有用と思うアプリを3.1節で説明したユーザ生成情報を利用して提示することができるかどうかを評価する。評価の基準として、ユーザが目的のアプリを見つけるまでに要したクエリの発行数と、目的のアプリが検索結果の何位に属していたかを利用する。

また、3.2節で提案した2つのクエリ入力方法と、3.3節で提案した2つのランキング方法について、それぞれの方法で結果がどのように異なるかを調査した。

5.1 実験用プロトタイプシステム

実験を行うために、第一段階で提案した手法に基づいて動作するアプリ検索システムのプロトタイプを実装した。このプロトタイプシステムはブラウザ上で利用するWebアプリとなっており、Perlで実装した。また、検索の対象とするアプリとして、AppStoreにてランキング上位に存在する日本語対応のアプリ100個を選び、あらかじめそれらのアプリの全ての紹介文、及びレビューがデータベースに保存されている。実装したプロトタイプシステムのスクリーンショットを図4に示す。

尚、プロトタイプシステムでは、tf-idfを求める際の idf_i の値がWeb全体を全ドキュメントとした時の値に近似できるとし、Web全体における単語 w を含むドキュメント数をYahoo!ウェブ検索Web API⁷⁾から取得して利用している。

5.2 実験内容

実験は被験者実験により行う。被験者には、前節で紹介したアプリ検索システムのプロトタイプシステムを利用してもらい、ある状況において有用なアプリの検索を行ってもらう。

(1) 被験者に対して、普段の生活で起こりうる状況の1シーンを提示する。提示する状況は、実験に用いた100個のアプリのうち、状況に応じた利用が考えられる10個のア

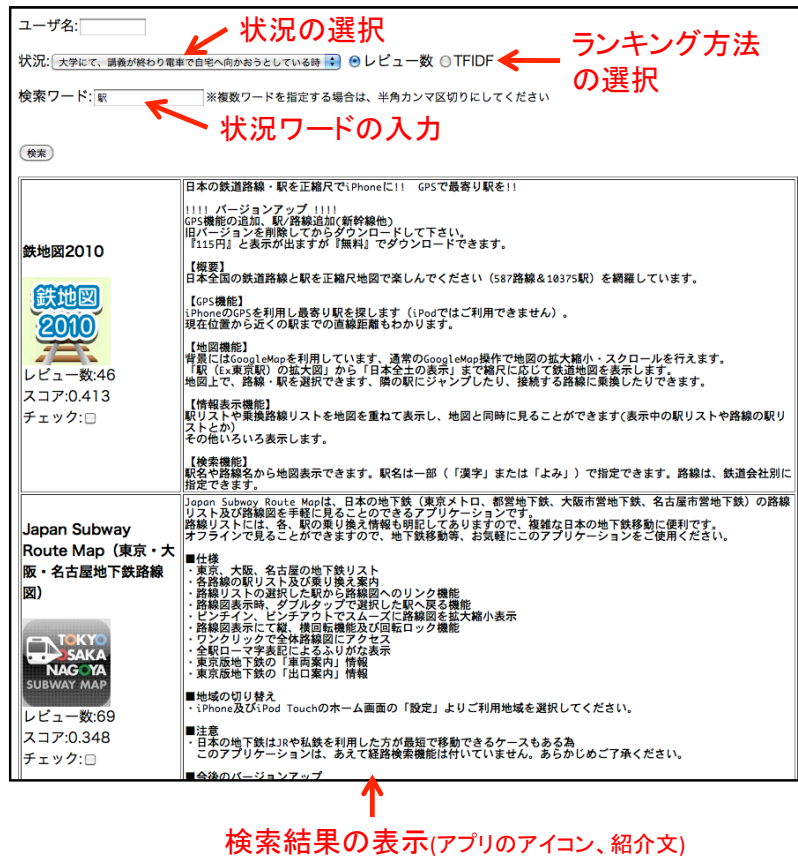


図 4 実験用に作成した検索システムのプロトタイプ (フリーワード入力の場合)
Fig. 4 Screenshot of the prototype system for the experiment

プリを選び、それぞれが有用になると思われる具体的な状況を文章で表現したものと
する。

- (2) 被験者には、自分が提示された状況にいる時の事を想像してもらい、プロトタイプシステムに自分の状況を表すキーワードをクエリとして入力してもらい。その際、半数の被験者に対しては 3.2.1 節の手法で、もう半数の被験者に対しては 3.2.2 節の手法により、検索クエリの入力を行ってもらい。
- (3) 3.3 節で示した 2 つのランキング手法によりアプリをランキングして提示し、それぞれのアプリの概要を確認してもらい、与えられた状況において有用なアプリだと思うものをチェックしてもらい。もしリスト中に該当するアプリがひとつも無ければ、新しい別のクエリを入力し、再度検索を行ってもらい。
- (4) 被験者が目的のアプリを得るまでに要した検索回数と、提示されたランキング中の順位を記録する。
- (5) (1)～(4) の作業をあらかじめ用意した 10 種類の状況に対して行ってもらい。

以上が実験の手順である。ここで、タグの選択によるクエリの入力では、あらかじめユーザに提示するタグのリストを用意しておく必要がある。このタグリストは、提示する状況の参考とした 10 個のアプリの全レビューの tf-idf を求めた結果の上位となったワードのうち、状況を表すものをいくつか手作業で選びだし、それらをまとめて順不同に並べたものとしている。

5.3 実験結果

フリーワード入力とタグ選択入力での各被験者の平均クエリ発行数を表 1 と表 2 に、提示されたランキング中、被験者が有用だと感じたアプリの順位の分布を図 5 と図 6 に示す。図 5 と図 6 で、「該当なし」とは、何度か検索を行っても該当するアプリが見つからず、被験者が検索を断念した場合である。

まず、図 5 と図 6 から、どの手法においても、ユーザが有用と感じるアプリが上位にランキングされており、アプリのレビューを用いることでユーザの入力した状況に適したアプリが推薦できていることがわかる。

次に、表 1 と表 2 から、平均のクエリ発行数はタグ選択入力で tf-idf の値によるランキングを行った場合が最も少なかった。これは、被験者に提示されたタグの候補はあらかじめ tf-idf の結果を元に生成されたものであるため、フリーワード入力の場合と異なり、選択したタグに対応するアプリが必ず存在し、微妙な言葉の表記の違いによりアプリが見つからない問題が無いためであると考えられる。また、フリーワード入力で tf-idf の値によるランキ

表 1 フリーワード入力での各被験者の平均クエリ発行数

Table 1 Average number of the query in free-word input

	レビュー	tf-idf
被験者 1	2.6	8.4
被験者 2	4.2	2.2
被験者 3	2.6	6.0
被験者 4	4.2	3.0
被験者 5	2.6	5.4
平均	3.2	5.0

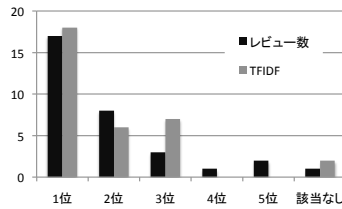


図 5 フリーワード入力での被験者が選んだアプリの順位分布

Fig. 5 Ranking distribution of the subject's selected applications in free-word input

ングを行った場合の平均クエリ発行数が最も多くなっているが、これは各アプリが被験者が入力したワードについて tf-idf の結果を持たない場合が多いからであったと考えられる。図 5 と図 6 から、被験者が選んだアプリの順位に関しては、各手法でそれほど大きな違いは見られなかった。

以上から、ユーザに自分の現在状況をクエリを入力してもらう場合は、フリーワード入力と比較してタグ選択入力の方が被験者にとって状況を入力しやすく、良好な検索結果が得やすいと考えられる。ランキング手法に関しては、クエリ入力手法との組み合わせによる相性はあるが、実際にどちらが有効かを確認する為には、より多くの被験者による調査が必要であると思われる。

6. 関連研究

ユーザの状況に応じた情報やサービスの推薦に関しては、これまでに幾つかの研究が行

表 2 タグ選択入力での各被験者の平均クエリ発行数

Table 2 Average number of the query in tag-select input

	レビュー	tf-idf
被験者 6	2.4	2.6
被験者 7	4.8	2.2
被験者 8	2.4	3.0
被験者 9	3.0	3.6
被験者 10	2.4	2.6
平均	3.0	2.8

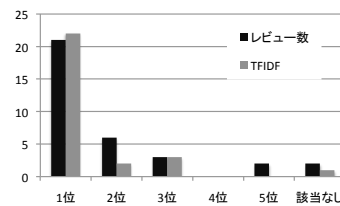


図 6 タグ選択入力での被験者が選んだアプリの順位分布

Fig. 6 Ranking distribution of the subject's selected applications in tag-select input

われている。奥ら⁸⁾は、ユーザの状況を複数のパラメータによる多次元ベクトルで表し、Support Vector Machine(SVM)を用いてユーザの嗜好と状況の関係性を学習することで、ユーザの状況に応じた嗜好の変化に対応した情報(店舗情報など)を推薦することを提案している。また、川原ら⁹⁾は、センサネットワークが存在する環境において、複数のセンサ情報とサービスの起動との関係性を統計的に学習し、もっとも起動されそうなサービスを推薦するプラットフォームを提案している。松本ら¹⁰⁾は、携帯電話におけるユーザの操作とコンテキスト情報との対応を学習し、あるコンテキスト情報においてよく利用されるアプリケーションを提示する手法を提案している。

我々の研究は、コンテキスト推定アルゴリズムに関してはこれまでに提案されてきた手法を参考に用いる予定だが、推定に必要な学習データの効率的な収集に着目している点の特徴である。これまでの研究は、推定アルゴリズムの提案と評価は様々に行われているが、あらかじめ十分な学習データが必要であり、その学習データの収集手法について考慮していないため、実際にサービスとして提供する上で問題があると考えられる。これに対し、我々の研究では、自分に必要なアプリを検索するという有用なサービスの提供を通じて、推定に必要な学習データを暗黙的に収集することができる。また、学習データは汎用的に利用可能なものであるため、アプリの推薦に限らず様々な状況依存型サービスの構築に利用することができる。

7. まとめと今後の課題

本研究では、ユーザにアプリ検索サービスを提供して学習情報を収集する段階と、収集した学習情報に基づいて状況を推定し推薦を行う段階の2つの段階により、ユーザの状況に応じてサービスを推薦するシステムの提案を行った。そして、提案したシステムの第1段階にあたる、ユーザの現在状況をクエリとするアプリ検索サービスについて、プロトタイプシステムの実装を行い、被験者実験を行った結果、ユーザの入力した状況に適したアプリが推薦できることが確認できた。

今後の課題として、今回行った実験はユーザに文章で状況を提示して検索を行ってもらうものであり、ユーザの実際の現在状況を入力するものではないため、実際の生活の中でアプリを検索してもらう被験者実験も行う必要があると言える。また、ランキングアルゴリズムのさらなる改良を行いたいと考えている。そして、このアプリ検索サービスを実際に多くの人に利用してもらい、学習データの収集を行った後、第2段階のシステムの実装と評価を行っていく予定である。

参 考 文 献

- 1) van Setten, M., Pokraev, S. and Koolwaaij, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS, Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Vol.LNCS3137, Springer Berlin/Heidelberg, pp.235-244 (2004).
 - 2) Konstan, J.A., Miller, B.N., Maltz, D., et al.: GroupLens: Applying collaborative filtering to Usenet news, Comm. ACM, Vol.40, No.3, pp.77-87 (1997).
 - 3) App Store, <http://www.apple.com/jp/iphone/apps-for-iphone> (2010-5-17 確認)
 - 4) Android Market, <http://www.android.com/market> (2010-5-17 確認)
 - 5) 矢野 幹樹, 岩崎 陽平, 河口 信夫: ”駅.Locky: 無線 LAN 位置推定を用いた時刻表アプリの開発”, 情報処理学会全国大会, 6ZP-4 (2010).
 - 6) 白木 敦夫, 矢野 幹樹, 松原 茂樹他: ”モバイルアプリケーション推薦のための Twitter 発言者の状況の推定”, マルチメディア, 分散, 協調とモバイル (DICOMO2010) シンポジウム (to appear).
 - 7) Yahoo!ウェブ検索 Web API, <http://developer.yahoo.co.jp/webapi/search/websearch/v1/websearch.html> (2010-5-17 確認)
 - 8) 奥 健太, 中島 伸介, 宮崎 純他: ”状況依存型ユーザ嗜好モデリングに基づく Context-Aware 情報推薦システム”, 情報処理学会論文誌:データベース, Vol.48, No.SIG11(TOD34), pp.162-176 (2007).
 - 9) 川原 圭博, 司 化, 森川 博之他: ”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, B-19-3 (2005).
 - 10) 松本 光弘, 清原 良三, 沼尾 正行他: ”携帯電話におけるユーザの操作パターンを用いたアプリケーション推薦方式の提案”, マルチメディア, 分散, 協調とモバイル (DICOMO2009) シンポジウム, pp.1149-1155 (2009).
-