

App.Locky: コンテキスト依存型サービス推薦を 目的としたユーザ状況収集プラットフォーム

矢野 幹樹^{†1} 梶 克彦^{†1} 河口 信夫^{†1}

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ものを探し出すことが困難となりつつある。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実現を目指す。推薦を実現するうえで必要となる学習情報の獲得を行うために、現在状況をクエリとするアプリ検索サービスを提供し、ユーザ状況の収集から推薦までを行うプラットフォームを提案する。提案に基づいた状況依存型アプリ検索システム「App.Locky」の実装を行い、インターネット上に公開したうえで、実ユーザを対象とした大規模検索ログ収集実験を行った。実験の結果、収集された検索ログをユーザ状況の推定に利用可能であることを確認した。

App.Locky: Users' Context Collecting Platform for the Context-aware Service Recommendation

MOTOKI YANO,^{†1} KATSUHIKO KAJI^{†1}
and NOBUO KAWAGUCHI^{†1}

Recently, it has become very hard for users to find their desired mobile services because the number of applications and Web services are rapidly increasing. Therefore, we aim at providing a system for recommending services according to the user's context. To collect learning information to estimate user's context, we propose a platform for collecting users' context and recommending services by providing an application search service that inquires user's context. We implemented a system named "App.Locky" based on our proposal and conducted experiments by publishing the system on the Internet. As a result, we confirmed that collected search logs can be used to estimate user's context.

1. はじめに

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できる環境が整いつつある。これにともない、モバイル端末向けに提供されるアプリケーション（以下アプリと呼ぶ）の数が爆発的に増加している。サービス数が増加するほどに、膨大な数のサービスの中からユーザが自分にとって必要なものを探し出すことは困難になりつつある。モバイル端末はユーザがいつでもどこでも持ち歩くものであるため、ユーザがおかれている状況の変化によって今現在必要なアプリも変化する。よって、モバイル端末向けアプリをユーザの状況に応じて推薦するシステムの実現には大きな意義がある^{*1}。

ユーザの状況に応じた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状況（位置、時間、プロフィールなど）との距離に基づく推薦¹⁾や、類似した状況のユーザが利用したサービスを推薦する協調フィルタリング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各サービスがどのようなコンテキスト（位置、時間など）において有用であるかという学習情報が大量に必要となり、学習情報が収集されていない場合は推薦を行えないという問題がある。たとえば App Store³⁾では40万本を超える iPhone アプリが提供されている。ある状況が与えられたときに各サービスが有用か否かを判定するには、各アプリについて様々な状況下でのアプリの有用性に関する情報が少なくとも10件程度は必要と考えられる。よって App Store に公開されている全アプリを対象とした推薦システムを構築するには、少なくとも400万件もの情報が事前に必要となる。また、現存するモバイル端末から自動で取得できる情報（以下コンテキスト情報と呼ぶ）としては、位置や時間のような基本的な情報しか獲得できないため、これらのコンテキスト情報では多様な状況をすべて表現することが困難である。たとえば「病気のため体調が悪い」や「忘れ物をした」というような状況は、ユーザの位置や時間というパラメータとは関連性の薄い事象であるため、位置や時間の情報から推測することは難しいと考えられる。

我々は、状況に応じて有効性が変化するサービスの例としてモバイル端末向けのアプリに

^{†1} 名古屋大学大学院工学研究科

Graduate School of Engineering, Nagoya University

^{*1} App Store における全 20 カテゴリの有料・無料アプリ上位 5 個を対象とし、状況に依存した利用が有効であるアプリと、状況に依存せず利用可能なアプリを人手で分類したところ、状況に応じた利用が有効であるアプリは約半数であった（2011年7月時点）。

着目し、ユーザの現在おかれている状況に適したモバイル端末向けアプリを自動推薦するシステムの実現を目標としている。既存手法における問題を解決するために、学習情報の不足や、コンテキスト情報の不足など、情報が不足している状態でも有用なサービスを提供し、多くの人々に利用してもらうことにより推薦に必要な情報を獲得し、自動推薦を行うサービスへとシームレスに発展させていくことのできるユーザ状況収集、およびサービス推薦プラットフォームを提案する。

提案するプラットフォームでは、ユーザが現在おかれている状況を表現するために「状況タグ」を導入する。状況タグは、「講義中」「飲み会」「空腹」「昼休み」といったように、ユーザの状況を自然言語で簡潔に表現したテキストであるため、人間が理解しやすいというメリットがある。状況タグ単体が表現する状況は単純なものであるが、複数の状況タグの組合せによって多様な状況を表現できる。

モバイル端末向けアプリの多くには、一般ユーザによるレビューが存在しており、自然言語で記述されたレビュー内には状況タグに相当するものが含まれている場合がある。そこで、アプリと状況タグとの関係を Web 上のアプリに関するユーザレビューから獲得する。本研究ではモバイル端末向けアプリ推薦を対象としているが、状況に応じて有効性が変化し、かつユーザレビューが存在しているようなサービスであれば本論文で述べるアプローチが適用可能である。

さらに、「現在状況をクエリとするアプリ検索サービス」を人々に利用してもらうことにより、コンテキスト情報と状況タグとの関係についてのデータを収集する。獲得された 2 つの関係を組み合わせ、統計的な学習により、コンテキスト情報から、ユーザが現在おかれていると思われる状況タグ集合を推測できるようになる。ここでアプリと状況タグの関係を利用すれば、その状況にあったアプリをユーザに推薦可能になる。

本論文では 2 章で関連研究を紹介し、それらの課題に対する我々のアプローチについて述べる。次に 3 章で、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より学習情報を収集するシステムの全体構成について述べる。4 章で、ユーザに現在状況の入力を求めた場合にどのような答えが得られるかを調査する予備実験を行い、5 章で、その考察をふまえて実装したアプリ検索システム「App.Locky」について述べる。6 章では、これを用いて行った実ユーザを対象とした大規模検索ログ収集実験とその結果について述べ、アプリ推薦システムへの移行可能性について検討する。最後に 7 章においてまとめと今後の課題を述べる。

2. 状況に応じたサービスの推薦

2.1 関連研究

ユーザの状況に応じたサービスの推薦に関する研究はこれまでに数多く行われている^{4),5)}。状況に応じたサービスの推薦を行うには、まず、ユーザのサービス利用の意思決定に影響を与える状況に関する情報（現在の位置や時間、興味や目的など）を取得（ユーザ状況の取得）し、次に取得したユーザ状況に基づいて、サービスの集合の中からユーザに推薦すべきサービスをフィルタリングし、提示する（情報フィルタリング）必要がある。

ユーザ状況の取得には、大きく分けて明示的手法と暗黙的手法の 2 種類がある。明示的手法では、現在の状況に関する情報をユーザに直接入力してもらう手法である⁶⁾。次に暗黙的手法では、端末のセンサ情報や操作履歴などをもとにユーザの状況を取得または推定する。SenSay⁷⁾では、携帯電話のマイク、3 軸の加速度センサ、可視光センサ、温度センサを利用してコンテキスト情報を取得し、あらかじめ定義したルールに基づいて、取り込み中、活動中、暇、ノーマルの 4 つの状況を判定し、それに応じて携帯電話のモードを変更するアプリケーションを実装している。

情報フィルタリングには、コンテンツの特徴に基づく手法と、協調フィルタリングを利用する手法の 2 つがある。サービスの特徴に基づく手法では、推薦対象のサービスに対して、それが有用となるコンテキスト情報の値を設定しておき、ユーザの状況に近いサービスを推薦する。林ら¹⁾は、コンテンツに対してあらかじめ推薦する条件となるコンテキスト情報のベクトル（加速度センサや温度センサ、GPS など複数のセンサの値のベクトル）を設定しておき、ユーザの端末から取得したコンテキストベクトルとコンテンツのベクトルとの距離を計算し、最も距離が小さいコンテンツをユーザに提示している。この手法では、コンテンツへの特徴付けを人手により行う手間が問題となるが、機械的に行う方法として、コンテンツの利用履歴などからそのコンテンツがよく利用される状況を統計的に学習する手法⁸⁾や、コンテンツ内のテキスト情報やコンテンツに対するレビューを利用する手法^{9),10)}も提案されている。また、協調フィルタリングは、あるユーザに対して情報を提示するとき、そのユーザと似た嗜好を持つ別のユーザの行動をもとに情報を取捨選択する技術である。これを利用してユーザに応じた商品の推薦を行うサービスとして Amazon がある¹¹⁾。

2.2 アプローチ

前述のように様々な既存手法があるなか、本論文ではサービスの中で特に現在爆発的に増加しているモバイル端末向けアプリケーション（以下アプリ）を対象とし、端末から暗黙的

に取得した情報に基づいてユーザの状況を推定し、自動的にアプリを推薦するシステムの実現を目指す。

モバイルアプリケーションの推薦に関する研究^{12),13)}も行われており、アプリの推薦に対しても、既存の手法は適用できると考えられる。しかし以下の点から、推薦システムを実際に利用可能で有用なサービスとして実現するうえでの課題があると考えている。

- アプリには飲食店情報や商品情報などの情報検索から、ゲームやツールなど多種多様なものが存在し、それらが有用となる状況も多様である。そのため、現在のモバイル端末から取得可能である位置や時間のような情報だけでは、多様な状況をすべて表現することが困難である。たとえば「病気のため体調が悪い」や「忘れ物をした」というような状況は、ユーザの位置や時間というパラメータとは関連性の薄い事象であるため、位置や時間の情報から推測することは難しいと考えられる。
- アプリの数は非常に多く、それらを既存の統計的手法によってフィルタリングするには各サービスがどのようなコンテキストにおいて有用であるかという学習情報が大量に必要になると思われる。たとえば App Store³⁾ では 40 万本を超える iPhone アプリが提供されている。ある状況が与えられたときに各サービスが有用か否かを判定するには、各アプリについて様々な状況下でのアプリの有用性に関する情報が少なくとも 10 件程度は必要と考えられる。よって App Store に公開されている全アプリを対象とした推薦システムを構築するには、少なくとも 400 万件もの情報が事前に必要となる。学習情報を収集するためには、多くのユーザに推薦システムを利用してもらう必要があるが、学習情報がないために有用な推薦結果を提示することができなければ、ユーザに推薦システムを積極的に利用してもらうことはできない。

上記の問題を解決するために、本論文では状況依存型サービス推薦システムに対して以下の要件を定義する。

- 現存するモバイル端末から容易に取得可能な位置情報、時間情報の 2 つをユーザ状況推定のために利用するコンテキスト情報と定義し、さらに、「病気で体調が悪い」や「忘れ物をした」といった、コンテキスト情報とは関連性の薄い状況にも対応できる。
- サービスと状況の関係についての学習情報が不足しており、ある状況が与えられたときにどのサービスが有用であるかを推定できない状態であっても^{*1}、ユーザにとって有用

*1 あるサービスについての学習情報が 2 件や 3 件のように少なくとも、学習によって分類器を構築することは可能であるが、そのサービスが有用な状況を網羅できておらず、推定精度は低くなる。一般に、誤りが少ないデータを多く用いて学習するほど推定精度は向上する。

なサービスを提供でき、それを多くのユーザに利用してもらうことによって、各サービスがどのようなコンテキストにおいて有用であるかという学習情報を大量に収集できる。

以下、これらの要件を満たすべく提案する、ユーザ状況収集およびサービス推薦プラットフォームについて述べる。

3. 提案システム

本論文では、前章であげた要件を満たす、ユーザ状況収集およびサービス推薦プラットフォームを提案する。本プラットフォームでは、学習情報の不足の課題を解決するために、各アプリの特徴を Web 上から獲得し、これを用いて「現在状況をクエリとするアプリ検索サービス」を提供しユーザに利用してもらうことで学習情報を収集する。この際、コンテキスト情報の不足の問題を解決するために「状況タグ」を利用する。

本章では、状況タグの利用と Web 上の既存情報からアプリの特徴を獲得する手法について述べ、提案するプラットフォームのシステム構成について述べる。

なお、本論文で提案するシステムでは、特にユーザが現在おかれている状況に着目し、ユーザの嗜好に関する情報は扱わないものとする。すでに多くの嗜好に基づく推薦システムが提案されており¹⁴⁾、それらの成果は提案システムに導入可能である。

3.1 状況タグによるコンテキスト情報の表現

前述したように、位置と時間などモバイル端末から暗黙的に取得可能なコンテキスト情報だけでは、アプリ利用の意思決定に影響を与えるユーザの状況をすべて表現することは難しい。また、アプリが有用となる状況を調べるにはユーザのアプリ利用履歴などを利用した統計的学習による手法が有効だと考えられるが、対象とするアプリの数が多い場合は膨大な学習情報が必要となる。

近年の Web サービスでは、映像や画像、音声のようなコンピュータによって意味を識別することが難しい情報に対して、人の手によって「タグ」と呼ばれるテキストによる付随情報を付与し、情報の検索を可能にする手法が用いられている。Web 上で写真を共有するサイトである Flickr¹⁵⁾ では、アップロードされた写真に対して、ユーザがその写真が撮影された場所の名前、写っている人の名前などの情報を自由に付与でき、キーワードによる写真の検索や関連した写真の検索などが可能である。

本論文では、上記の手法を参考にし、アプリが有用となる状況を「状況タグ」によって表現する。状況タグは、「講義中」「飲み会」「空腹」「昼休み」といったように、ユーザの状況

を自然言語で簡潔に表現したテキストである。状況タグ単体が表現する状況は単純なものであるが、複数の状況タグを組み合わせることによって複雑な状況を表現できる。

状況タグは自然言語によって作られるため、コンテキスト情報だけでは表現できない状況を扱える。そして、コンテキスト情報とアプリとの関係を状況タグにより媒介することで、アプリと状況タグ、状況タグとコンテキスト情報との関係をそれぞれ別の手段で獲得することができ、段階的な学習情報の収集を可能にする。また、状況を推定するためのコンテキスト情報が不十分である場合でも、そのコンテキスト情報から推定される複数の状況タグ(誤ったものを含む)をユーザに提示し、正しい状況タグだけを選別してもらうことができる。また、選別された結果を状況タグの推定精度を向上させるためのフィードバックとして利用できる。

各サービスがどのような状況で利用可能であるかに関するオントロジを規定する手法も存在するが^(16),17)、状況タグはオントロジを規定する場合にも有効に利用できる。アプリ提供者が網羅的にオントロジに関する情報も提供しているとは限らず、またユーザレビューからもオントロジを規定するための情報を取得できるとは限らない。よってオントロジはどのような状況であれば利用可能であったというデータをもとにボトムアップに構築することになる。アプリと状況タグの関係に関する情報を収集し解析すれば、それはオントロジのボトムアップな生成の補助となるだろう。

3.2 システム構成

本節では、提案するプラットフォームのシステム構成について述べる。本システムは2つの段階に分かれている。第1段階では、Web上のアプリに関するユーザレビューを利用して獲得した状況タグとアプリとの関係に基づいたアプリ検索サービスを提供し、この運用から状況タグとコンテキスト情報との関係を学習するための検索ログを収集する。検索ログとは、コンテキスト情報と、そのときユーザが現在の状況として入力した状況タグが関連付けられた記録である。前節で述べたWeb上のユーザレビューを利用することで状況タグとアプリの関連情報が取得でき、本節での検索ログによってコンテキスト情報と状況タグの関係が取得できるようになる。第2段階では、これらの情報を用い、コンテキスト情報からそのコンテキストに該当すると思われる状況タグを統計的に推定し、推定された状況タグに関連するアプリを自動的に推薦する。第1段階の概要を図1に、第2段階の概要を図2に示す。

3.2.1 第1段階: 現在状況をクエリとするアプリ検索サービス

第1段階では、ユーザに対して「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、ユーザに今自分がどのような状況にいるのか

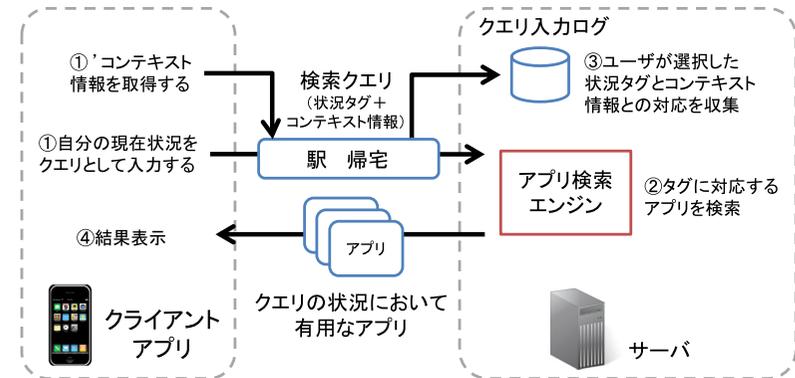


図1 第1段階: 自分の現在状況をクエリとするアプリ検索サービス
Fig. 1 1st stage: Application search service based on the context of input by the user.

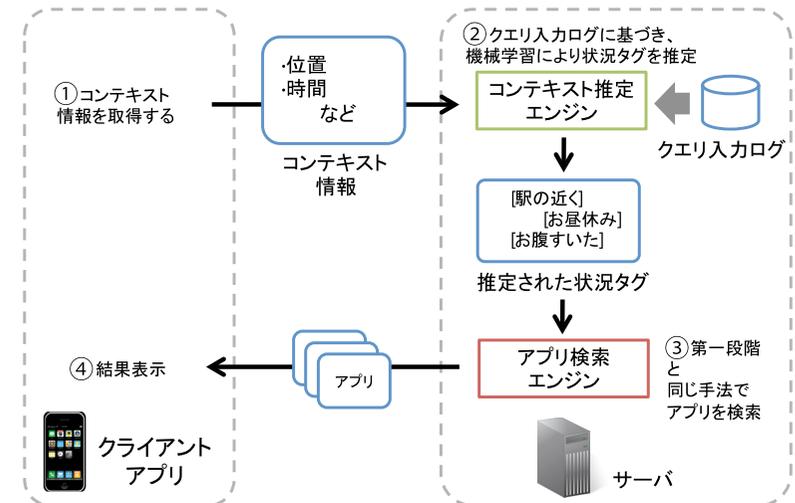


図2 第2段階: 検索ログに基づく状況依存型アプリ推薦
Fig. 2 2nd stage: Context-aware application retrieval/recommendation based on the search log.

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サービスである。ユーザ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、データベースに蓄積する。これにより、状況タグとコンテキスト情報との相関を統計的に獲得することができる。

ユーザが入力した状況からアプリを検索するための情報としては、関連研究でコンテンツの特徴を取得する手法としてあげた、Web上に存在するレビューを利用する。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 Market¹⁸⁾などでは、アプリに対してユーザがレビューを投稿できるサービスが提供されている。また、Web上には、ブログなどにアプリのレビューを掲載するユーザが多数存在する。このような文章中では、アプリが有用となる状況に関する記述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で投稿されたもの)を以下に示す。駅.Lockyは、位置依存型時刻表アプリであり、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¹⁹⁾。

- 「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おり、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ことができる。この性質を利用し、レビュー中から入力された状況に関する語がどの程度含まれているかに基づいてアプリのランク付けを行い、検索結果を提示できると考えられる。

3.2.2 第2段階: 検索ログに基づく状況依存型アプリ推薦

第2段階では、コンテキスト情報からその状況において入力される状況タグを自動的に推定する。そして、第1段階と同じ手法によりアプリの推薦を行う。第1段階で取得した検索ログ(モバイル端末から自動取得されるコンテキスト情報と、そのときユーザが選択した状況タグのログ)を学習データとした統計的学習を行い、過去の似た状況でのユーザが最も多く入力した状況タグと見なす。ユーザはこれにより第1段階のように自分から状況タグを入力する必要なしに、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でない状態であっても、アプリ検索サービスとしてユーザに有用なサービスを提供

することができる。そして、アプリ検索サービスによって学習情報が収集され、状況の自動推定が可能になった時点から、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。

4. 予備実験

提案するシステムの実装を行う前に、ユーザが自分の現在状況を表す状況タグの入力を求められた場合に、どのような答えが入力されるかを調査する予備実験を行った。

4.1 実験概要

本実験では、Webサイト検索エンジンのような形式で、Web上でアプリを検索するプロトタイプシステムを実装し、被験者に検索キーワードとして現在状況の入力を求める。このプロトタイプシステムをWeb上で一般公開して被験者を募集し、アプリの検索を行ってもらった。この際、被験者の募集にはiPhone向け時刻表アプリ「駅.Locky」を利用した。駅.Lockyは本実験を行った2010年6月時点で約25万ダウンロードされており、通勤時に毎日利用するアクティブユーザも多い。そのため、このアプリを通じて実験への協力を呼びかけることで、多くのユーザからの実験結果を収集できると考えられる。駅.Lockyで最もユーザが目にする画面において、「情報スペース」と呼ばれる情報表示領域を設けて、本実験への協力の呼びかけを行った。情報スペースのスクリーンショットを図3に、フリーワード入力で状況タグを入力しアプリの検索を行う画面のスクリーンショットを図4に示す。実験は2010年6月22日から7月2日までの10日間行い、この期間中にユーザから入力された状況タグをすべて検索ログとして記録した。

4.2 予備実験の結果と考察

実験結果の概要を表1に示す。駅.Locky上での募集効果もあり、実験期間の10日間での計3,146件の検索ログが収集された。また、入力された状況タグの種類は770種類であり、重複する内容の状況タグが複数の被験者から入力されていることが分かる。

そのような重複している状況タグの数を集計し、時間帯別で上位順にまとめた結果を表2に示す。この結果から、朝は「出勤」、昼は「ランチ」、夕方以降は「帰宅」、夜は「寝る」などのタグが上位に来ており、時刻に応じたタグの傾向がよく現れていることが分かる。しかし、今回の実験は「駅.Locky」という時刻表アプリ上で行ったものであるため、多くのユーザは駅.Lockyを駅や電車の乗車中などに利用する機会が多く、得られたクエリは電車に関係したものに偏ってしまっていた。

次に、入力されたクエリがどの程度重複しているかを調べるために、重複数が n 以上で



図 3 駅.Locky 情報スペース上での実験協力の呼びかけ
Fig. 3 Call for collaboration for the experiment on Eki.Locky information space.



図 4 駅.Locky 上で行った実験サイトのスクリーンショット
Fig. 4 Screenshot of the web site for the experiment on Eki.Locky.

表 1 予備実験でのデータ収集状況

Table 1 Status of data-collection in the preliminary experiment.

| | |
|------------------------|---------------|
| 実験開始日時 | 2010年6月22日14時 |
| 実験終了日時 | 2010年7月2日14時 |
| 集計期間 | 10日間 |
| 実験リンクへのアクセス数 | 4,714回 |
| 検索回数 | 3,146回 |
| 入力されたクエリの種類 | 770種類 |
| 被験者が App Store へ移動した回数 | 463回 |

あるクエリが、全 770 種類のクエリの中で占める割合をプロットしたグラフを図 5 に示す。この図から、全体の 80% の状況タグが重複のないものだと分かる。検索ログから状況タグとコンテキスト情報との相関を抽出するためには、同じ状況タグが入力された際の複数のコンテキスト情報が必要であるが、本実験の結果では重複した状況タグのクエリがほとんど存在しないため、相関を抽出するのに十分ではないと考えられる。入力されたクエリの中には、「家に帰る」と「帰宅」や、「電車の中」と「電車内」というように、ほぼ同じ状況を意味するが表現が異なる状況タグが多く見受けられた。

最後に、本実験で入力されたクエリのうち、その内容が状況タグの形式に不適当なもの(「自宅」と「テレビ」ではなく、「自宅でテレビを見てる」のように、文章になっているも

表 2 時間帯別上位状況タグと入力回数

Table 2 The 5 most frequently inputted tags in 3 hours.

| 0~3時 | | 3~6時 | | 6~9時 | | 9~12時 | |
|------|----|------|---|------|----|-------|----|
| 寝る | 10 | 出勤 | 4 | 出勤 | 19 | 出勤 | 11 |
| 眠い | 7 | 帰宅 | 2 | 通勤 | 19 | 通勤 | 7 |
| 自宅 | 5 | 通勤 | 2 | 出社 | 7 | 電車 | 6 |
| 電車 | 4 | 朝 | 2 | 自宅 | 7 | 出版 | 4 |
| 寝る前 | 2 | | | トイレ | 4 | 暇 | 4 |

| 12~15時 | | 15~18時 | | 18~21時 | | 21~0時 | |
|--------|---|--------|----|--------|----|-------|----|
| 帰宅 | 7 | 帰宅 | 19 | 帰宅 | 66 | 帰宅 | 44 |
| 電車 | 6 | 電車待ち | 3 | 帰宅中 | 14 | 電車 | 8 |
| 仕事中 | 3 | 仕事 | 3 | 電車 | 6 | 寝る | 7 |
| トイレ | 3 | 通学 | 3 | 乗り換え | 5 | 眠い | 7 |
| ランチ | 3 | 電車 | 3 | テレビ | 5 | サッカー | 6 |

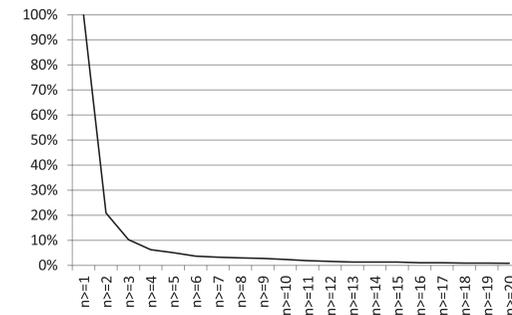


図 5 重複数が n 以上であるクエリが全 770 種類のクエリの中で占める割合

Fig. 5 Percentage of the queries which were inputted over n times in all 770 collected queries.

のや、適当に入力され言葉になっていないもの)がどの程度混入しているかを調査した。その結果、適さないクエリは 115 個あり、全体の 15%であった。このため、フリーワード入力でユーザに状況タグの入力を求めたとき、状況タグとして適切でない内容が入力される場合が一定の割合で存在するため、そういったノイズを除去する操作を行う必要があることが分かった。

本予備実験の結果から、ユーザが入力する状況タグには、時刻や時期など、コンテキスト情報との相関が現れており、コンテキスト情報から状況タグを推定するための学習情報として検索ログを利用することの有効性がある程度確認できた。しかし、ユーザからフリーワー

ド入力で状況タグの入力を求める場合の問題点がいくつか判明した．それらの問題点をもとに，ユーザに状況タグの入力を求めるうえでの要件を以下にあげる．

- (1) 駅などの特定の場所に限られない，幅広い環境下で検索を行ってもらう．
- (2) 同じ状況を意味するが，異なる表現の状況タグをまとめる．
- (3) 状況タグとして適切でないクエリを除去する，もしくは適切な形式に変換する．

次章では，これらの要件に基づいて開発した状況依存型アプリ検索システム「App.Locky」の実装について述べる．

5. 状況依存型アプリ検索システム「App.Locky」の実装

本章では，3章で提案したシステムの構成と，4章の予備実験で得られた知見をもとに実装した状況依存型アプリ検索システム「App.Locky」について述べる．App.Locky は iPhone 上で動作するクライアントアプリケーションと，サーバによって構成されている．クライアントアプリ側でユーザに自分の状況に関するタグを選択してもらい，サーバ側で選択された状況タグに関係したアプリを検索し，ユーザに提示する．

5.1 システム構成

App.Locky のシステム全体の構成を図 6 に示す．App.Locky はユーザが現在状況の入力とアプリ検索結果の閲覧を行うクライアントアプリと，入力された状況タグをもとに，アプリ情報データベースからアプリを検索し，結果をクライアントアプリへと送信する検索サーバによって構成される．クライアントアプリは iPhone/iPod touch 専用の OS である iOS 上で動作する．検索サーバは Perl と MySQL によって実装されており，CentOS 上で動作している．

ユーザが本システムを用いてアプリを検索する場合の動作の流れを以下に述べる．

- (1) クライアントは検索サーバに質問および状況タグの情報を要求する．
- (2) 検索サーバは，「今どこにいる？」「今なにしてる？」などの状況に関する質問と，その答えとなる状況タグ候補の状況タグツリー（5.2 節参照）の情報をクライアント側に送信する．
- (3) 質問・タグ情報を受信したクライアントは，その情報に基づいて状況タグの候補をタグ選択ユーザインタフェース上に提示し，状況タグの入力を求める．質問の提示と状況タグの選択は 1 度の検索で複数回行われる．
- (4) すべての質問への回答が完了すると，クライアントはクエリジェネレータで，選択されたタグの情報と，コンテキストマネージャで生成されたユーザのコンテキスト情報

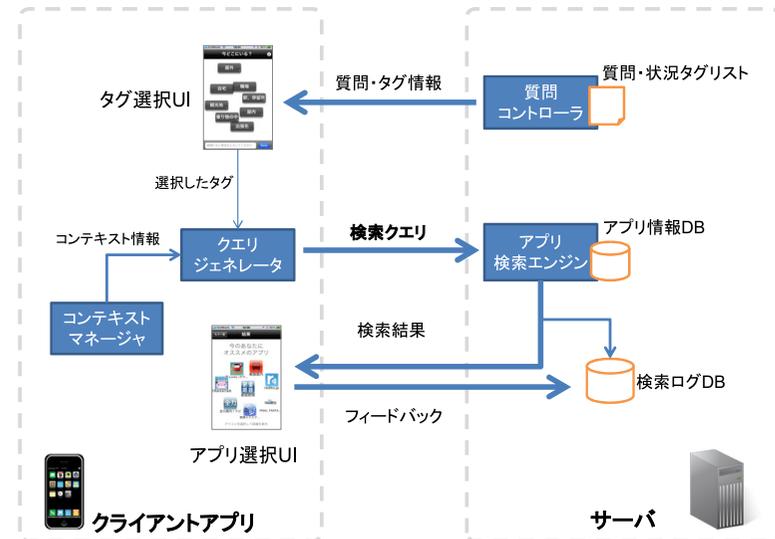


図 6 App.Locky のシステム構成
Fig. 6 System architecture of App.Locky.

を結合し，検索クエリを生成した後，検索サーバへと送信する．

- (5) 検索クエリを受信した検索サーバは，アプリ検索エンジンで，アプリのリストと各アプリのレビュー情報を保持するアプリ情報データベースの情報をもとに検索クエリに対応するアプリを検索し，ユーザ側に提示するアプリの一覧を作成する．この際，ユーザから送信された検索クエリは，検索サーバの検索ログデータベースに保存される．
- (6) 検索結果を受信したクライアントは，アプリ選択ユーザインタフェース上に結果のアプリをユーザに提示する．ユーザはこの中から，気になったアプリの詳細を確認したうえで，必要だと思ったアプリを選択し，端末へとアプリをダウンロードする．もし詳細を表示したアプリが必要ないと感じた場合には「このアプリは必要ない」というボタンを押してもらう．ここでユーザの行った操作はすべてサーバ側へと送信され，この情報も検索クエリと同じく検索ログデータベースへと記録される．

このように，App.Locky は提案したシステムの第 1 段階「現在状況をクエリとするアプリ検索サービス」を iPhone 上で実際に利用可能なアプリケーションとして実装したものとなっている．

5.2 状況タグツリー選択インターフェイス

従来より、画面上に表示されたオントロジやタグを選択することにより、それをクエリとして Web ページやアプリケーションを検索するというシステムが存在している^{20);21)}。これらの手法では、クエリ入力にかかる手間を削減し、フリーワード入力による表記ゆれが起こるのを防ぐことができるため、4 章で判明したフリーワードによる状況の入力に関する問題を解決することができる。App.Locky は「状況タグツリー」を利用し、これらの従来研究と同様、ユーザに自分の状況を表現するタグの選択を求める。状況タグツリーとは、人の生活上で一般的に起こりうる状況を表す状況タグを、抽象的な状況から具体的な状況への結び付きにより表現したツリー構造である。ユーザはあらかじめ用意された状況タグツリーのツリー構造をたどり、自分の状況を表現する状況タグを 1 つないし複数探し、選択する。あらかじめ用意された候補の中から選択を求めることで、同じ状況でも状況タグの表現が異なるという問題や、状況タグとして不適切な言葉が入力されるという問題を未然に防ぐことができる。また、抽象的な状況から具体的な状況へとたどることで、目的の状況タグを容易に探索できると考えられる。さらに、ユーザが意識していなかった自身の状況を、状況タグの候補から気づくことができるメリットも考えられる。

我々は、語彙間の階層関係として日本語語彙体系²²⁾ や Yahoo!カテゴリ²³⁾ のツリー構造を部分的に参考にしつつ、この状況タグツリーの作成を手作業で行った。その際、最上位の階層で「今どこにいる?」「今なにしてる?」などの状況の基底となる要素に関連付けて分類した。作成した状況タグツリーの一部を図 7 に示す。また、各分類とそれに属する状況タグの数を表 3 に示す。

これに加えて、App.Locky ではユーザに「今」に関係した状況タグの選択を意識させ、モバイル端末上で直感的かつ容易に状況タグツリーからの状況タグの探索、選択が可能となるように、ばねモデルを用いたインタラクティブな状況タグ選択インターフェイスになってい

表 3 ユーザに提示される質問の内容と状況タグの数
Table 3 Questions and numbers of tags to show to the users.

| 分類 | 状況タグ数 |
|-----------|-------|
| 今どこにいる? | 73 |
| 今なにしてる? | 129 |
| 今の気分は? | 20 |
| 今の体調は? | 11 |
| あなたはどんな人? | 27 |
| 今どんな時? | 0 |

る。本インターフェイスのスクリーンショットを図 8, 図 9 に示す。本インターフェイスでは、状況タグツリーの特定の階層における状況タグがそれぞれ長方形のプレート上に表示され、各プレートの位置がばねモデルに基づいてリアルタイムに変化する。ユーザはリアルタイ

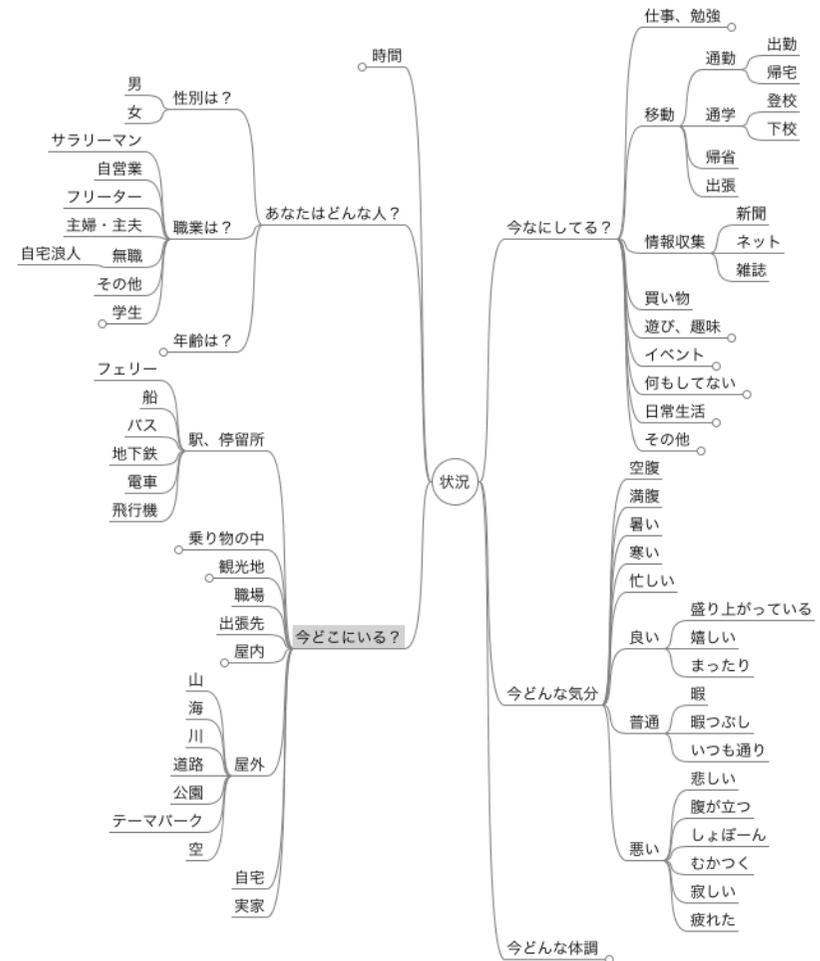


図 7 作成した状況タグツリーの一部
Fig. 7 A part of context-tag tree.



図 8 状況タグ選択画面

Fig.8 Screenshot of tag selecting view.



図 9 検索結果画面

Fig.9 Screenshot of result view.

ムに移動するプレートの中から、自分の状況に一致する状況タグが書かれたものを選択する。もし選択した状況タグが内部ノード（下位の状況タグを子に持つ状況タグ）であった場合は、同階層の他のタグが消え、その状況タグが画面中心へと移動した後、選択した状況タグを中心として下位の状況タグが新しく画面上に出現する。このようにして、上位階層から下位階層へと移動しつつ、より具体的な状況を表す状況タグを選択することが可能である。

5.3 検索アルゴリズム

App.Locky では、入力した状況タグと同じ語がレビュー中にどの程度含まれるかを考慮して、アプリの検索スコアを計算する。ここでは、まず文書検索などで頻りに用いられる tfidf を用いる。tfidf では、多くのドキュメントに出現する語は重要度が下がり、少ないドキュメントにしか現れない語の重要度を上げることができる。ただし、複数の状況タグがクエリとして入力された場合、クエリに含まれる特定の状況タグの重要度が高すぎると、その状況タグの影響によって特定のアプリが上位にランキングされてしまうことがある。入力された複数の状況タグによって現在のユーザの状況を表現していることを考えると、特定のタグのみの影響が大きくなるのは好ましくない。そこで、ここでは、入力された状況タグの語のレビュー中での tfidf スコア（式 (2)）に加え、複数提示された質問のうち、何個の質問をカバーしているかの質問カバースコア（式 (5)）を組み合わせて利用する。

本システムにおけるアプリ a の状況タグ集合 T に対する検索スコア $Score(a, T)$ の計算式を以下に示す。

$$R_a = \{r_i\} \quad (1)$$

$$tfidf(t, a) = tf_{t, R_a} \cdot idf_t \quad (2)$$

$$tf_{t, R_a} = \log \frac{n_{t, R_a}}{\sum_k n_{w_k, R_a}} \quad (3)$$

$$idf_t = \log \frac{|R|}{|\{r_i : t \in r_i\}|} \quad (4)$$

$$f(t, a) = \begin{cases} 1 & \exists i : (t \in r_i) \cap (r_i \in R_a) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$Score(a, T) = \sum_{t \in T} f(t, a) + tfidf(t, a) \quad (6)$$

ここで、 n_{t, R_a} はあるアプリ a のレビュー集合 R_a 中のタグ t の出現頻度、 $\sum_k n_{w_k, R_a}$ は、 R_a に含まれる全単語の出現頻度の合計、 $|R|$ は全アプリの総レビュー数、 $|\{r_i : t \in r_i\}|$ はタグ t を含むレビュー数である。また、 R_a は 1 つのアプリのレビュー r_a の集合、 T は全質問に対して選ばれた状況タグ t の集合を表す。クライアントから T が与えられたとき、サーバは式 (6) を用いて全アプリに対して $Score(a, T)$ を求め、スコア上位約 10 個のアプリを検索結果としてクライアント側に送信する。

6. 検索ログ収集実験とその結果

本章では、不特定多数のユーザから実生活におけるアプリ検索ログを収集するために、本章で実装した「App.Locky」を日本国内の iPhone ユーザに対して公開し、実ユーザを対象として実施した大規模検索ログ収集実験について述べる。実験を通して、App.Locky によってコンテキスト情報と状況タグの関係に関する検索ログを大量に収集できることを確認し、また、コンテキスト情報と状況タグ、状況タグどうし、状況タグとアプリの間に相関を見出しうるかについて検証することで、第 2 段階の状況依存型アプリ推薦への移行可能性を示す。

本実験では、App Store に提供されている全 20 カテゴリの有料・無料アプリ上位 100 個ずつを、実験期間中に毎日取得したものを対象とした。実験期間中に検索対象となったアプリ数は約 10,000 個であった。

6.1 実ユーザを対象とした大規模検索ログ収集実験

コンテキスト情報から状況タグの推定を行うためには、収集された検索ログ中のコンテ

キスト情報と状況タグとの関係に何らかの統計的特徴が現れている必要がある。そこで、App.Locky を日本国内の iPhone ユーザに対して公開し、多くの人々に実際に利用してもらいつつ検索ログを収集し、統計的学習を行ううえで十分な学習情報が収集可能であるかを確認する。また、検索ログ中に傾向がどのように現れているかを、状況タグと時間、状況タグと位置、状況タグどうしの関係、状況タグとアプリのそれぞれの関係について調査し、コンテキスト情報から状況タグを統計的学習により推定可能であるかを検証する。

我々は App.Locky のクライアントアプリを 2010 年 12 月 23 日より App Store 上に公開した。さらに、より多くのユーザに利用してもらうために、4.1 節の予備実験で行ったのと同様に、駅.Locky 上の情報スペース上でアプリの紹介を行った。

6.2 検索ログの分析とその結果

検索ログの統計的性質を検証するために行った調査内容とその結果を述べる。集計期間は 2010 年 12 月 24 日から 2011 年 1 月 24 日である。この期間中に、App.Locky は 6,500 人以上のユーザにダウンロードされた。

6.2.1 App.Locky 利用状況

App Store 上に App.Locky を公開して以降の本アプリの利用状況について、収集された検索ログの概要を表 4 に、時刻・曜日別の検索回数の分布を図 10 に示す。

表 4 検索ログの概要
Table 4 Overview of search log.

| | |
|-----------------------|--------|
| App.Locky ダウンロード数 | 6,557 |
| 総検索回数 | 21,382 |
| 「今の状況」での検索回数*1 | 15,130 |
| 位置情報ありの検索回数(今の状況)*2 | 11,565 |
| 検索ユーザ数(個人情報送信者のみ)*3 | 4,139 |
| アプリの詳細を表示した回数 | 19,628 |
| 詳細が表示されたアプリの種類 | 1,721 |
| 「このアプリを入手」が選ばれた回数 | 2,793 |
| 「このアプリを入手」が選ばれたアプリの種類 | 633 |
| 「このアプリは不要」が選ばれた回数 | 1,136 |

*1 App.Locky では今の自分に関係のない状況を入力することもできるよう、入力した状況タグが「今の状況」に即したものであるかどうかを確認する仕組みになっている。

*2 端末で取得した位置情報の送信の有無を任意に設定できるため、すべての検索ログに位置情報が含まれてはいない。

*3 位置情報と同様に端末識別番号の送信の有無もユーザが任意に設定できるため、個人情報送信を許可したユーザ数のみカウントが可能。

App.Locky は駅.Locky 上で紹介した効果もあり非常に多くのユーザに利用してもらうことができた。また、検索を行ったユーザの多くが「今の状況」での検索を行い、位置情報も含めた検索クエリの送信を行ったユーザも多かった。位置情報を利用した有用なアプリが広く普及したことにより、ユーザの位置情報の送信に対する抵抗が小さくなっていることがこの要因として考えられる。

表 4 から、15,000 回以上の検索で提示された結果に対して、「このアプリを入手」が選択された回数は全体の 20%程度であった。App.Locky でアプリを検索し、アプリを入手するというプロセスを、Web ページに表示された広告をクリックし、その商品を購入するプロセスと見なすと、広告をクリックしたユーザが商品を購入する割合(コンバージョン率)は一般には 15%程度あれば高いと見なされており、それと比較しても本実験でのアプリ入手割合は高いといえる。App.Locky はある程度ユーザの求めるアプリを提示することができたといえる。また、App.Locky はアプリを検索することを目的としたアプリであるため、基本的にユーザはアプリを入手したいと考えており、検索結果を好意的に受け止めてくれるユーザが多かったものと考えられる。

ただし、約 10,000 個のアプリが検索対象となっていたが、実際に入手されたアプリの種類は 1 割にも満たなかった。人気のある上位 100 件のアプリであっても、そのアプリに対するレビューが少ないものがあるため、状況タグとアプリを適切に関連付けることができなかったと考えられる。また、そもそも状況への依存性の低いアプリも数多く存在していることから、このように少ない割合のアプリしか入手されなかったのだと考えられる。

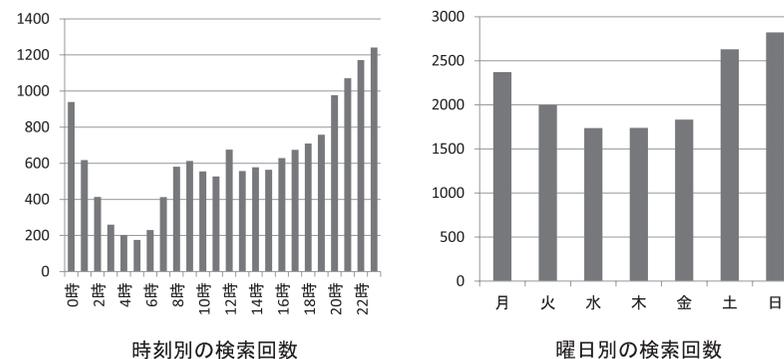


図 10 時刻・曜日別の検索回数の分布
Fig. 10 Distribution of search count by hour and weekday.

6.2.2 状況タグ選択状況

検索ログ中の状況タグの選択状況を調査する。

まず、選択された状況タグの上位 10 個と、状況タグ選択回数の合計、そのうちフリーワードで入力されたユーザ生成タグの数を各質問ごとにまとめたものを表 5 に示す。ここで、選択回数の合計が質問ごとに異なるのは、「今どこにいる?」「今何してる?」以外の質問が検索のたびにランダムに変化する仕組みにしているためである。

次に、各状況タグがそれぞれ何回選択されているのかを調査した。選択回数が n 以上である状況タグの、全状況タグに対する割合を質問ごとにまとめたものを図 11 に示す。

この図から、250 種類以上あるタグの多くが 10 回以上は選択されており、予備実験の結果と比較して、各状況タグについて統計的解析をするために十分な検索ログを収集することができ、状況タグツリー選択インタフェースの有効性を示していると思われる。しかし、表 5 にあるとおり、「自宅」などの一般的にユーザが iPhone を利用する状況についてタグの選択回数が集中する傾向が現れていた。タグの選択をより分散させるためには、本アプリを自宅や職場以外の状況においても積極的に利用してもらうような施策を施す必要があると考えられる。

なお、我々が作成した状況タグツリーには含まれていない、フリーワードで入力された

表 5 選択された状況タグの質問別上位 5 件とその選択回数
Table 5 Top 5 most selected tags and the number of each selected count.

| 今どこにいる? | | 今なにしてる? | | 今の体調は? | |
|---------|--------|---------|--------|---------|-------|
| 自宅 | 7,035 | 暇 | 3,746 | 普通 | 2,962 |
| 電車 | 2,067 | ネット | 1,159 | 眠い | 1,782 |
| 職場 | 1,358 | 何もしてない | 926 | 疲れた | 1,659 |
| 実家 | 1,099 | 帰宅 | 763 | 元気 | 1,106 |
| 出張先 | 313 | 睡眠 | 539 | 良い | 476 |
| 合計 | 15,102 | 合計 | 15,102 | 合計 | 9,191 |
| ユーザ生成タグ | 376 | ユーザ生成タグ | 698 | ユーザ生成タグ | 219 |

| 今の気分は? | | あなたはどんな人? | |
|---------|-------|-----------|-------|
| 寒い | 1,366 | 男 | 332 |
| 満腹 | 688 | サラリーマン | 131 |
| 空腹 | 598 | 30 代 | 104 |
| 忙しい | 438 | 20 代 | 91 |
| 暇つぶし | 293 | 女 | 75 |
| 合計 | 5,478 | 合計 | 1,309 |
| ユーザ生成タグ | 144 | ユーザ生成タグ | 58 |

ユーザ生成タグの情報も表 5 に示すように数多く収集されており、これを用いて状況タグツリーをさらに拡張できると思われる。

6.2.3 状況タグの時間依存性

時刻に応じて状況タグの選択にどのような傾向が現れるかを調査する。表 6 に、時間帯別の上位 5 つの状況タグを示す。そして、個々の状況タグの時間依存性を、ある時刻 T において、あるタグ x が選択される確率 $P(x|T)$ を求め、選択回数が多いものから選んだ 2 つ

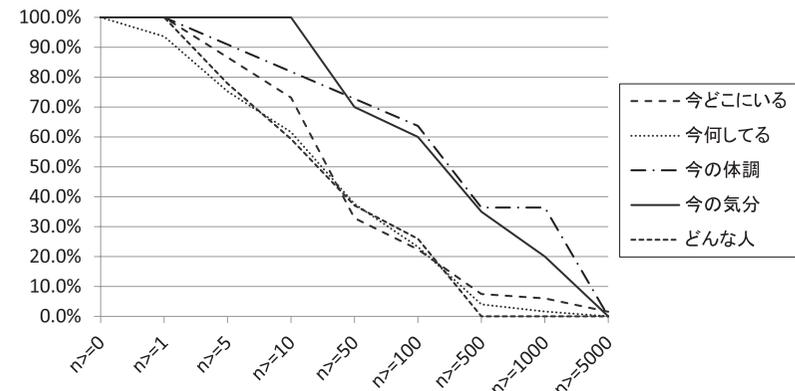


図 11 選択回数が n 以上である状況タグの全状況タグに対する割合
Fig. 11 Percentage of the context tags which were selected over n times in all collected context tags.

表 6 時間帯別上位状況タグ (値は選択された回数)
Table 6 Top 5 most selected tags in 3 hours and the number of each selected count.

| 0~3 時 | | 3~6 時 | | 6~9 時 | | 9~12 時 | |
|-------|-------|-------|-----|-------|-----|--------|-----|
| 自宅 | 1,311 | 自宅 | 390 | 自宅 | 500 | 自宅 | 662 |
| 暇 | 594 | 暇 | 176 | 電車 | 291 | 普通 | 422 |
| 眠い | 419 | 眠い | 155 | 普通 | 269 | 暇 | 409 |
| 普通 | 341 | 普通 | 112 | 暇 | 255 | 電車 | 298 |
| 疲れた | 268 | 実家 | 75 | 出勤 | 221 | 職場 | 215 |

| 12~15 時 | | 15~18 時 | | 18~21 時 | | 21~0 時 | |
|---------|-----|---------|-----|---------|-----|--------|-------|
| 自宅 | 590 | 自宅 | 682 | 自宅 | 947 | 自宅 | 1,968 |
| 暇 | 471 | 暇 | 496 | 暇 | 571 | 暇 | 887 |
| 普通 | 406 | 普通 | 432 | 普通 | 523 | 普通 | 700 |
| 職場 | 365 | 職場 | 255 | 電車 | 494 | 疲れた | 505 |
| 電車 | 213 | 電車 | 254 | 疲れた | 341 | 電車 | 420 |

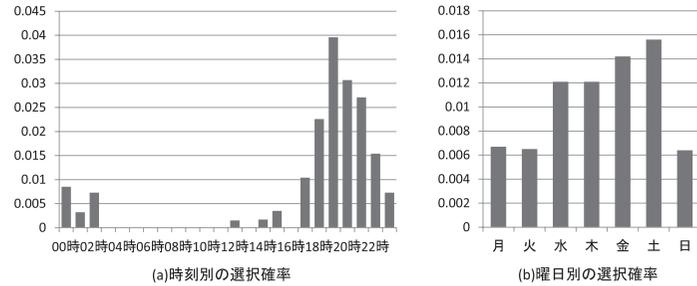


図 12 「夕食」の時刻・曜日別選択確率

Fig. 12 The probability of a tag: "Dinner" selected by hour or weekday.

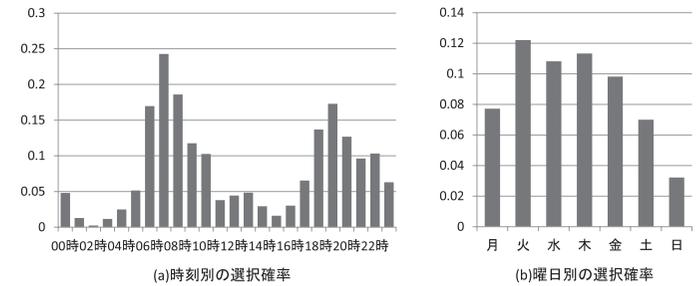


図 13 「通勤」の時刻・曜日別選択確率

Fig. 13 The probability of a tag: "Commuting" selected by hour or weekday.

の状況タグの時刻・曜日別の選択確率を図 12 と図 13 のグラフ上に示す。

これらの図から、各状況タグの時間・曜日別の選択確率にはその特徴がよく現れていることが分かり、この確率の情報から、時間のコンテキスト情報に基づいた状況タグの推定も可能であると考えられる。

また、今回の調査では行っていないが、ある状況タグ A をすでに選択している場合に、ある状況タグ B が選択される確率の時間推移にも大きな特徴が現れる可能性があると考えられる。たとえば、「睡眠」という状況タグの時間別確率の分布は、「20 代」というタグが選択されている場合と、「60 代」というタグが選択されている場合とで異なる可能性があるため、今後はそういった視点での特徴の調査も必要である。

6.2.4 状況タグの位置依存性

位置に応じて状況タグの選択にどのような傾向が現れるかを調査する。図 14 は、東京都

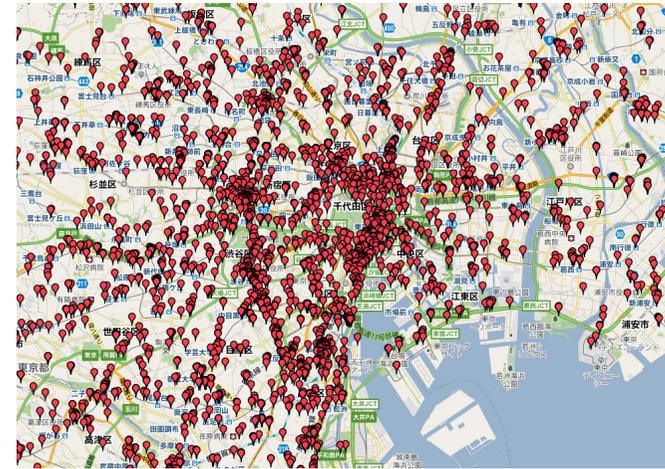


図 14 東京都内での検索位置の分布

Fig. 14 Distribution of locations where users has conducted a search.

内での検索位置の分布をマッピングしたものである。次に、特定の位置周辺における状況タグの傾向を調査するために、特定の施設付近での状況タグの分布を調査した。ここでは、日本全国の鉄道の駅の周囲約 300 m 以内での各状況タグの選択回数を検索ログから集計し、状況タグ x が選ばれたときに、それが駅周辺 L で選択された場合の条件付き確率 $P(L|x)$ を計算した。なお、選択された回数が 10 以下である状況タグ x は、統計的な傾向が十分に現れていないものと考えて除外した。 $P(L|x)$ の高い順にタグを並べた結果(質問別)を表 7 に示す。

この結果から、駅周辺においては駅に関係した状況タグが選択される確率が高くなっていることが分かり、状況タグと位置についても特徴が現れているものと思われる。今回は調査の対象として駅周辺のみを扱ったが、今後はショッピングモールや繁華街、学校などの周辺においても特徴の調査が必要である。

6.2.5 状況タグの共起関係

App.Locky では 1 回の検索で複数の質問が提示されるため、ユーザは 1 回の検索で複数の状況タグを選択する。そのため、同時に選択された状況タグの間には何らかの共起関係が現れる可能性がある。ここでは、選択回数の多かったいくつかの状況タグについて共起関係の調査を行う。

表 7 駅周辺 300 m 以内での選択率が高い状況タグ上位 5 件 (質問別) と $P(L|x)$ の値

Table 7 Top 5 tags which are most likely to be selected around stations within 300 m and the number of $P(L|x)$ each.

| 今どこにいる？ | | 今なにしてる？ | | 今の体調は？ | |
|---------|-------|---------|-------|--------|-------|
| 山 | 0.824 | 寝る | 0.833 | 疲れた | 0.583 |
| 駅, 停留所 | 0.773 | 鉄道 (趣味) | 0.828 | 普通 | 0.577 |
| 地下鉄 | 0.772 | 帰宅 | 0.759 | 元気 | 0.573 |
| 電車 | 0.770 | 出勤 | 0.752 | 腹痛 | 0.567 |
| 駅 | 0.769 | 下校 | 0.750 | 眠い | 0.560 |

| 今の気分は？ | | あなたはどんな人？ | |
|--------|-------|-----------|-------|
| 暑い | 0.669 | 40 代 | 0.759 |
| 空腹 | 0.650 | 60 代 | 0.692 |
| 病気 | 0.619 | サラリーマン | 0.664 |
| いつも通り | 0.602 | 30 代 | 0.635 |
| 満腹 | 0.601 | 男 | 0.598 |

表 8 共起度の高い状況タグ

Table 8 Top 5 high co-occurrence tags.

| 電車 | 職場 | 買い物 |
|----|--------|-----------|
| 登校 | 休憩 | ショッピングモール |
| 帰宅 | ミーティング | デパート |
| 下校 | 仕事 | コンビニ |
| 出勤 | 営業中 | ショップ |
| 帰省 | バイト | 60 代 |

共起の強さを数値化する指標としては, 次式 (7) で表される相互情報量を用いて 2 つの状況タグ x, y の共起度 $I(x, y)$ を求めた.

$$I(x, y) = \log \frac{p(x, y)}{P(x)P(y)} \quad (7)$$

ここで, タグ「 x 」とタグ「 y 」の単独での選択確率をそれぞれ $P(x), P(y)$, 2 つのタグが同時に選択された確率を $P(x, y)$ とする. 表 8 に, 選択回数の多かった状況タグの中から選んだ「電車」「職場」「買い物」の 3 つのタグについて, 共起度の高いタグ上位 10 個をまとめた. 表中の値は $I(x, y)$ を表している.

表 8 から, 状況タグどうしの共起関係には分かりやすく特徴が現れていることが分かる. よってこの情報を利用し, ユーザが最初に 1 つの状況タグを選択した時点で次に選択する状況タグを推定し提示することが可能になると思われる. また, たとえば「あなたはどんな

表 9 アプリに対する共起度の高い状況タグ

Table 9 Top 5 high co-occurrence tags with applications.

| 食べログ | NAVITIME | ウェザーニュース |
|---------------------------------|-------------------------------------|--------------------------|
| 現在地周辺の飲食店情報を検索するアプリ (選択回数: 160) | 乗換案内や経路探索などを行うナビゲーションアプリ (選択回数: 87) | 天気予報を見るためのアプリ (選択回数: 78) |
| 昼食 1.891 | 駅, 停留所 3.564 | 60 代 4.043 |
| しょぼーん 1.269 | 自転車 3.148 | サッカー 3.043 |
| 飲食店 1.007 | ギャンブル 3.148 | アウトドア 2.821 |
| 食事 0.784 | 飛行機 2.770 | 海 2.458 |
| 盛り上がっている 0.684 | 出張 1.916 | 暑い 1.628 |

人？」に関する状況タグは, コンテキスト情報に応じて変化することのない静的なものであるため, そういった静的な状況タグはあらかじめ選択されたものと仮定したうえで, それと関連の強い状況タグを最初から推定して提示することも可能だと思われる.

6.2.6 アプリと状況タグの関係

検索結果からユーザが自分の状況で使えるものだと判断されたアプリと, そのときにユーザが選択していた状況タグとの関係を調査する. ここでは, ユーザにより選ばれた回数の多かったアプリを取り上げ, そのアプリをユーザが取得した際の共起度の高い状況タグを調べた. 共起の強さは, 前項と同じように相互情報量を用い, アプリ a と状況タグ x の共起度 $R(a, x)$ を求めた. 表 9 に, 検索結果中から選択された回数の多かった 3 つのアプリについて, アプリの名前とそのアプリの概要, 共起度の高いタグ上位 5 個をまとめた. 表中の値は $R(a, x)$ を表している.

この結果から, ユーザが取得したアプリと共起度の高い状況タグには, ある程度そのアプリの性質に関係したものが多く含まれる傾向が現れているといえる. ただし, App.Locky では, 選択された状況タグから 5.3 節で示した検索アルゴリズムに基づいて選ばれた 8 つのアプリをユーザに提示し, ユーザはその中からしかアプリを選ぶことができないため, アプリと状況タグの共起度は必然的に検索アルゴリズムの影響を大きく受けていると考えられる. したがって今回の調査では, 実際にそのアプリが有用な状況の状況タグとの共起度が必ずしも高くなっているわけではない可能性がある. また, アプリ「ウェザーニュース」において「60 代」というタグの共起度が高いのは, 60 代がウェザーニュースのアプリを入手した回数が少なかったが, 検索ログ全体で「60 代」のタグが選ばれた回数も少なく, 相対的に共起度が向上したためと考えられる.

ユーザからの適切なフィードバックを得るためには, もし検索結果中に有用となるアプリ

が存在しなかった場合に、上位 8 位以下の他のアプリも探すことができるようにする必要がある。また、もしユーザ側がその状況で有用なアプリを知っている場合には、そのアプリを教えてもらうというフィードバックの獲得の方法も考えられる。

6.3 得られた知見

本実験では、駅.Locky 上でのプロモーションの効果もあり多くの人々に App.Locky を利用してもらうことができた。アプリ検索という有用なサービスを提供し、多くのユーザに利用されることで大量の検索ログを収集できたといえる。

収集された検索ログから、コンテキスト情報と状況タグとの相関や、状況タグどうしの相関がみられる組合せの存在が確認された。そのため、時間や位置といったコンテキスト情報から一部の状況タグの推定は可能といえる。また、状況タグに基づくアプリ検索によって提示されたアプリが 20%程度の割合で入手されたことを考えると、一部の状況であれば、コンテキスト情報から状況タグを推定し、その状況タグに基づいてアプリを推薦できるといえる。ただし、すべての状況タグについてコンテキスト情報やアプリとの特徴的な相関がみられたわけではないため、自動的に取得するコンテキスト情報を増やしたり、第 2 段階における状況タグを推定するアルゴリズムの工夫などが必要であると考えられる。

4 章で行った予備実験と本実験との比較では、まず、図 5 と図 11 の比較から分かるように、本実験では選択された回数の多い状況タグの割合が増加している。予備実験のフリーワード入力では 1 つの状況タグが重複して入力される確率が低かったが、提案した状況タグツリー選択インタフェースによって異なる状況タグの統一や、適切でない状況タグの除外がうまく行えていることが分かる。次に、表 2 と表 6 の比較から、本実験では予備実験であまり入力されていなかった「自宅」の状況タグが最もよく入力されている結果となっており、より実生活に即した検索ログの収集ができた。以上から、実装した App.Locky によって予備実験での課題を解決できたと考えられる。ただし、実生活に即した検索ログが得られたとはいえ、「自宅」など一部の状況タグにユーザの選択が集中してしまう傾向があり、今後はさらに多様な状況下での本アプリの利用を促進していく必要があると分かった。

7. まとめと今後の課題

本論文では、ユーザの状況に応じたサービスの推薦を目的とし、推薦に必要な学習情報を収集しつつサービスの推薦を行うプラットフォームを提案した。そして、この提案に基づきアプリ検索システム「App.Locky」の実装を行い、App Store に公開し不特定多数のユーザに実生活の中で利用してもらい、検索ログを収集する実験を行った。結果、6,500 人

以上のユーザが本システムを利用し、21,000 件の検索ログが収集された。この検索ログの分析により、App.Locky が提示したアプリのうち、20%程度のアプリが入手されたことが分かった。また、コンテキスト情報と状況タグの相関、状況タグどうしの相関、状況タグとアプリとの相関を統計的に抽出できることが確認された。これにより、コンテキスト情報が与えられたときに一部の状況を推定でき、状況タグが与えられたときにその状況に適したアプリを提示可能であることが示されたため、これらを組み合わせることによって、一部の状況においてはコンテキスト情報からのアプリの自動推薦が実現可能といえるだろう。

今後は、得られた検索ログに基づいて状況タグを推定するアルゴリズムの検討を行い、App.Locky へと実装したうえで、今回の実験と同様に実サービスとして利用してもらうことで推定結果の評価を行っていきたいと考えている。また、入力された状況においてより有用で、かつそのユーザが利用したことのないアプリを提示できるように検索アルゴリズムの改良も進める必要がある。そして、今回収集された情報から得られた知識モデルをアプリ以外の推薦にも適用し、幅広く状況依存型サービス推薦を発展させてゆきたいと考えている。

参考文献

- 1) 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: 情報ネットワーク, Vol.104, No.691, pp.149-154 (2005).
- 2) 奥 健太, 中島伸介, 宮崎 純ほか: 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).
- 3) Apple Inc.: App Store (online), available from (<http://www.apple.com/jp/iphone/apps-for-iphone>) (accessed 2011-03-15).
- 4) 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.3137, pp.515-548 (2004).
- 5) Gregory, A., Christopher, A., Jason, H., et al.: Cyberguide: A mobile context-aware tour guide, *Wireless Networks*, Vol.3, pp.421-433 (1997).
- 6) 梶 克彦, 平田圭二, 長尾 確: 状況と嗜好に関するアノテーションに基づくオンライン楽曲推薦システム, 情報処理学会音楽情報科学研究会報告, 2004(127), pp.33-38 (2004).
- 7) Siewiorek, D., Smailagic, A., Furukawa, J., et al.: SenSay: A Context-Aware Mobile Phone, *Proc. 7th IEEE International Symposium on Wearable Computers, ISWC*

- '03, IEEE Computer Society, p.248 (2003).
- 8) 川原圭博, 司 化, 森川博之ほか: 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).
 - 9) Niklas, J., Hagen, W.S., Christoph, M.M., et al.: Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations, *Proc. 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion, TSA '09*, ACM, pp.57-64 (2009).
 - 10) 杉木健二, 松原茂樹: カスタマーレビューに基づく商品検索システム, 楽天研究開発シンポジウム 2008, pp.45-48 (2008).
 - 11) Linden, G., Smith, B. and York, J.: Amazon.com recommendations: Item-to-item collaborative filtering, *Internet Computing*, Vol.7, No.1, pp.76-80, IEEE (2003).
 - 12) Woerndl, W., Schueller, C. and Wojtech, R.: A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications, *22nd International Conference on Data Engineering Workshops*, pp.871-878 (2007).
 - 13) 松本光弘, 清原良三, 沼尾正行ほか: 位置情報を含む携帯端末利用履歴からのコンテキストに基づく最適アプリケーション推定法の提案, 情報処理学会研究報告 MPS, Vol.2010-MPS-80, No.3, pp.1-8 (2010).
 - 14) 土方嘉徳: 嗜好抽出と情報推薦技術, 情報処理学会誌, Vol.48, No.9, pp.957-965 (2007).
 - 15) Yahoo! Inc.: Flickr (online), available from <http://www.flickr.com/> (accessed 2011-03-15).
 - 16) Sasajima, M., Kitamura, Y., Naganuma, T., et al.: Task ontology-based modeling framework for navigation of mobile Internet services, *Proc. IASTED European Conference: Internet and Multimedia Systems and Applications*, pp.47-55 (2007).
 - 17) Xia, H. and Yoshida, T.: Web Service Recommendation with Ontology-based Similarity Measure, *Proc. 2nd International Conference on Innovative Computing, Information and Control*, p.412 (2007).
 - 18) Google Inc.: Android Market (online), available from <http://www.android.com/market> (accessed 2011-03-15).
 - 19) 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, 情報処理学会第 72 回全国大会, pp.6ZP-4 (2010).
 - 20) Mirizzi, R., Ragone, A., Di Noia, T. and Di Sciascio, E.: Semantic wonder cloud: Exploratory search in DBpedia, *Proc. 10th International Conference on Current Trends in Web Engineering (ICWE '10)*, pp.138-149 (2010).
 - 21) Hassan-Montero, Y. and Herrero-Solana, V.: Improving Tag-Clouds as Visual Information Retrieval Interfaces, *InScit2006: International Conference on Multidis-*

ciplinary Information Sciences and Technologies (2006).

22) 池原 悟, 宮崎正弘, 白井 諭ほか: 日本語語彙体系, 岩波書店 (1997).

23) Yahoo Japan Corporation: Yahoo!カテゴリ (オンライン),
入手先(<http://dir.yahoo.co.jp/>) (参照 2011-03-15)

(平成 23 年 3 月 25 日受付)

(平成 23 年 9 月 12 日採録)



矢野 幹樹

2009 年名古屋大学工学部電気電子・情報工学科卒業。2011 年同大学大学院工学研究科博士前期課程修了。ユビキタスコンピューティングに関する研究に従事。



梶 克彦 (正会員)

2002 年名古屋大学工学部電気電子工学科卒業。2007 年同大学大学院情報科学研究科博士課程修了。博士 (情報科学)。同年 NTT コミュニケーション科学基礎研究所リサーチアソシエイトを経て, 2010 年より, 名古屋大学大学院工学研究科助教。人工知能学会, 日本ソフトウェア科学会, ヒューマンインタフェース学会各会員。実世界コミュニケーション, Web コンテンツ処理, 音楽情報処理に関する研究に従事。



河口 信夫 (正会員)

1990 年名古屋大学工学部電気電子工学科卒業。1995 年同大学大学院工学研究科情報工学専攻博士課程満了。同年同大学工学部助手, 同大学講師, 准教授を経て, 2009 年より同大学大学院工学研究科教授。モバイルコミュニケーション, ユビキタスコンピューティング, 行動センシングの研究に従事。博士 (工学)。ACM, IEEE, 人工知能学会, 日本ソフトウェア科学会, 電子情報通信学会, 日本音響学会各会員