

# 実世界行動のための Web 検索手法の提案

前川 卓也<sup>†</sup> 柳沢 豊<sup>†</sup> 岸野 泰恵<sup>†</sup> 亀井 剛次<sup>†</sup> 櫻井 保志<sup>†</sup>  
岡留 剛<sup>†</sup>

<sup>†</sup> NTT コミュニケーション科学基礎研究所  
京都府相楽郡精華町光台 2-4

E-mail: <sup>†</sup>{maekawa,yutaka,yasue,yutaka,yasushi}@cslab.kecl.ntt.co.jp, <sup>††</sup>houmi@idea.brl.ntt.co.jp

あらまし 本稿では、ユビキタス・センサ環境におけるクエリフリー web サーチ手法を提案する。この検索手法は、ユーザが行っているモノを使った行動に関する web ページを、モノに添付したセンサノードから得たデータを用いて自動的に検索する。例えばユーザがコーヒーメーカーを洗っているときに、「コーヒーメーカーを洗うときは、酢を使えば汚れがよく落ちる」といった、行動に係る知識やノウハウを含む Web ページを取得することを目的とする。キーワード ユビキタスコンピューティング, Web 検索, センサ, ADL

## A Proposal on Web Search method for ADLs

Takuya MAEKAWA<sup>†</sup>, Yutaka YANAGISAWA<sup>†</sup>, Yutaka YANAGISAWA<sup>†</sup>, Yutaka  
YANAGISAWA<sup>†</sup>, Yutaka YANAGISAWA<sup>†</sup>, and Takeshi OKADOME<sup>†</sup>

<sup>†</sup> NTT Communication Science Laboratories  
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

E-mail: <sup>†</sup>{maekawa,yutaka,yasue,yutaka,yasushi}@cslab.kecl.ntt.co.jp, <sup>††</sup>houmi@idea.brl.ntt.co.jp

**Abstract** This paper proposes a query free web search method in ubiquitous sensor environments. The method automatically retrieves a web page which relates to an user's activity by using sensor data obtained from sensors which are attached to physical objects. When washing a coffee maker, for example, we aim to retrieve an activity related web page including such information as 'cleaning a coffee maker with vinegar removes its stain well.'

**Key words** Ubiquitous computing, Web search, Sensor, ADL

### 1. ま え が き

生活に関わる身の回りの家電や家具がインターネットにつながる時代がやってこようとしている。そのような時代、web ページは PC のディスプレイに表示されるだけではなくなるだろう。すでに近年の多くのテレビは LAN ポートを備えており、画面に web ページを表示できるテレビも登場している。インターネットに接続したキッチン家電（冷蔵庫や電子レンジ）は、その小さな画面にページの一部やサマライズしたページを表示することが可能になるだろう。また、近年普及しているインターネットに接続した音楽プレイヤーが web ページを音声読み上げすることも可能だろう。そして、このような生活に溶け込んだデバイスにより人々は日常生活に関する追加情報やアドバイスを含むページを享受できるようになるだろう。人は、行っている行動に関する情報により利益を享受することが多い。例えば、人が髭をそっているときに、「髭をそる最適な時刻は起床後 10 分である」という情報を含むページが提示されればその人にとって有用だろう。

しかし、上記のような生活に溶け込んだデバイスは、PC とは異なりキーボードのような高速にクエリを入力するためのインタフェースをもたない。また、クエリを入力するためには、行っている日常行動を中断しなければならない。このような問題を解決する有効な方法は、ユーザが行っている行動に関するクエリを何らかの方法で自動作成し、そのクエリにマッチするページを提示する方法だろう。クエリを自動で生成し、それに合致するページを検索することをクエリフリーサーチ [7] と呼ぶ。上記のような環境でクエリフリーサーチを実現するためには、ユーザが行っている行動を検知する必要があるだろう。

一方、近年モバイルコンピューティング技術、センシング技術の進展により、小型の無線センサノードを安価で大量に生産できるようになってきている。センサノードにはノードの周囲を観測するセンサや、ノード自身に起こった現象を観測するセンサなどがインストールされており、環境や人の動きの観測に用いられることが多い。本研究では、センサノードをカップ、歯ブラシ、髭剃りといった日常物に添付し、その動きをリアルタイムに取得することを想定する。そして、ある時区間に使わ

れている（動かされている）モノの名前からクエリを自動生成し、そのクエリに合致するページを検索エンジンから取得することで、その時区間に行われた行動に関する web ページを得る（以降ではユーザによる行動を Activity of Daily Living の略である ADL と呼ぶ。）例えば、ある時区間に cup と milk と cocoa が使われて（動かされて）いれば、“cup milk cocoa” といったクエリを生成し、そのクエリに合致するページを取得する。これは、行動に関係するモノの名前を含むページは、その行動に関係しているだろうという我々の考えを基にしている。また、モノそのものに関する情報を含むページも得られると考える（例えば cocoa の健康効果に関するページなど）。ただし、われわれが目指しているのは生活行動に関する追加情報などを自動で検索することで、ユーザが所望の情報を自動で検索することではない。

ADL に関するページを提示することで下記のような利益がもたらされると考える。(1) 前述のコーヒーマーカーと酢の例のように、ADL をさらに洗練させるような情報をユーザにもたらしすることができる。(2) トリビアに関する書籍やテレビ番組の人気ぶりから分かるように、多くの人は世の中の事物や現象に関する背景知識を得ることで満足を感じる。提案手法により ADL や ADL に用いられるモノに関する背景知識を提供できる。(3) ADL に関する（新）製品情報を効果的にユーザに提供できる。ユーザが ADL を行っている際に、その ADL に関する興味深い製品の情報が提示されれば購買につながるだろう。

そして、本提案を達成するためには下記の 2 つの主要な問題を解決しなければならない。

-混同の問題: ユーザが行っている行動に直接関係のないモノが使われる（動かされる）ことがある。例えば、ユーザがお茶をいれるためにお茶や砂糖などを棚から取り出しているときに、関係のないココアや緑茶の容器を動かすことは日常生活では当たり前に行われるだろう。また、環境に居る複数のユーザが異なる行動を同時に行っているとき（例えば 1 人は歯ブラシで歯を磨いており、あと 1 人が髭剃りで髭をそっている。）、そのユーザらが使っているモノは近いタイミングで動かされるため混同するだろう。そこで、行われている行動ごとにモノをクラスタリングする必要があるだろう。

-クエリ作成の問題: 我々はクラスタごとにクラスタに含まれるモノの名前を用いてクエリを作成する。しかし、モノの名前のみを含むクエリはその表現があいまいである。例えば、“cup green-tea” といったあいまいなクエリで有用な情報を取得するのは難しいだろう。

また、上記のクラスタリングの段階で関係のないモノがクラスタにノイズとして含まれてしまうことは避けられない。つまり、クラスタに含まれるモノからシンプルにクエリを作成したら、クエリの AND 条件によりノイズを必ず含むページしか取得できない。正しくクラスタリングされたとしても、クラスタが含む全てのモノを語としてクエリを作成すればクエリをシビアにしすぎることがある（例えばクラスタに 10 個のモノが含まれる場合など）。

本論文の貢献は、将来のユビキタス環境における日常行動のための web 検索というコンセプトの提案と、上記問題を解決するための手法の提案である。上記の混同の問題を解決するために、モノ同士が使われた時刻間の距離、モノ同士のセマンティックな関連性、モノが使われた過去履歴などを用いて、同じ行

動に関係するモノのみからなるクラスタを作成する。クエリ作成の問題においては、あいまいなクエリを具体化するためにクエリ拡張により具体化を行う。また、クラスタのノイズやシビアすぎるクエリに対処するため、クラスタからある程度の数のモノ（語）をピックアップして複数のサブクエリを作成する手法を適用する。ピックアップすることで、ノイズを含まないサブクエリやシビアすぎないサブクエリをいくらかは作成できる。そして、複数のサブクエリを検索エンジンに送信することで得られた web ページ（複数のランキング）を、クラスタとの関連性を測る指標でリランクする手法を適用する。

本論文の構成は以下の通りである。2 章において研究背景について述べる。3 章で手法の提案を行い、4 章で関連研究との比較を行い、5 章で結論を述べる。

## 2. 背景

### 2.1 センサノード

本研究は、センサノードを屋内のモノに添付することを想定する。ここでは、われわれが実装したセンサノードを用いる。実装したセンサノードは加速度センサを搭載しており、約 30Hz のサンプリングレートで 3 軸加速度を計測し、サーバに無線通信で送信する。われわれが加速度センサを利用した理由は以下の通りである。(1) 赤外センサなどのように指向性がないため添付したモノの位置に依存せずにデータを取得できる。(2) エンターテインメント用、デジタル機器用に近年多用されており、最もメジャーなセンサの 1 つになりつつある。つまり、今後より一層の低価格化や小型化が期待される。(3) モノが使われている区間はモノを用いた人の行動に関する最もシンプルで根本的な指標であるため、それを検知できる加速度センサを用いた。この区間は、RFID、接触センサ、傾きセンサなどの他の多くのセンサでも取得可能である。

### 2.2 モノの利用の定義

加速度センサはあるレートでサンプルした加速度を数値として連続的に出力する。本研究では加速度の振幅が大きく変化している区間をモノが使われている（動いている）区間とする。以降ではその区間を変化区間と呼ぶ。われわれは、3 軸 (x, y, z) 加速度センサを用いているため、それぞれの軸のセンサデータから抽出された変化区間の和集合の区間を、そのセンサノードが添付しているモノが利用された区間とする。

信号処理の分野では、変化区間の検出に学習を用いることが多い。例えば [17] では、あらかじめ変化区間と雑音区間（変化区間以外の区間）のフーリエ成分を Gaussian Mixture Model (GMM) により学習することで変化区間を検出している。本研究でもほぼ同様のアプローチを用いる（この処理はセンサデータ処理の問題であるため、詳細な説明は省く。）

## 3. アプローチ

図 1 に、ADL のための web 検索のアーキテクチャを示す。本手法では  $n$  分ごとにその時点から過去  $n$  分間の変化区間をノードごとに求める。そして、求めた変化区間を用いて web 検索を行う。提案するアーキテクチャは主に 2 つのコンポーネントに分かれている。まず、 $n$  分間において一緒に使われているモノからなるクラスタを作成する。そして、クラスタごとにクラスタに含まれるモノの名前を用いて web を検索し、クラスタに対応する web ページを求める（クラスタごとに 1 つのペー

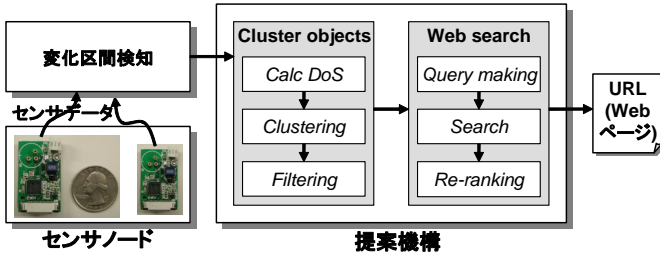


図1 ADLのためのWeb検索のアーキテクチャ。

ジが得られる)。われわれは  $n = 3$  とした。その理由として以下を挙げる。事前実験において、複数のモノを使った作業時間が約2分程度だったため、3分の時区間から得られる情報で十分の精度が得られると考えた。また、ユーザの行動を邪魔しないためにも、3分程度の間隔が適当と考えた。

以下では、“Cluster objects”について3.1節で、“Web search”について3.2節で説明する。

### 3.1 Cluster objects

図1に示したように、このプロセスは以下に示す3つの手順からなる。ここで、ある1つの行動に使われたモノ(1対1)同士の指標を以下では‘Degree of being used in the Same ADL (DoS)’と呼ぶ。モノ間のDoSが大きければモノ同士が同じ行動において使われた‘可能性’が高いとする。まず、ある区間に使われたモノ同士のDoSを計算し(Calc DoS)、DoSにしたがってモノをクラスタリングする(Clustering)。その区間において短時間しか使われていないモノのみを含むクラスタをフィルタし、残りを出力する(Filtering)。以下にそれぞれの手順を説明する。

#### 3.1.1 Calc DoS

DoSの計算手法を説明する。手法は主に下記の3つの指標を基にモノXとYのDoSを計算する。

-  $Temp(X, Y, t)$ : 時区間  $t$  において、XとYが近い時間に使われたかどうかの程度を表す。XとYの時区間  $t$  における変化区間を用いる。近い時刻に使われたモノは同じADLに使われている可能性が高いだろうという考えを基にしている。

-  $Hist(X, Y)$ : 過去において、XとYが同時に使われていたかどうかの程度を表す。つまり、あらかじめある程度の期間のデータセットを用意しておく。過去の行動において一緒に使われていたモノは同じADLにつかわれる可能性が高いだろうという考えを基にしている。

-  $Sem(X, Y)$ : XとYが意味的に近いかどうかの程度を表す。検索エンジンのヒット数を用いて計算したXとYの共起を用いる。実世界において一緒に使われるモノは、実世界を反映したWWWの文書中でも共起して現れるだろうという考えを基にしている。

以上の3つの指標を用いてDoSは下記のように表される。

$$DoS(X, Y, t) = Temp(X, Y, t) \cdot Hist(X, Y) \cdot Sem(X, Y).$$

では3つの指標について詳しく説明する。

#### [Temp(X, Y, t)]

Tempの計算方法を説明する前に、用語を図2を用いて説明する。 $x_i$ はXの、 $y_j$ はYの変化区間をそれぞれ表す。 $x_i$ はXの*i*番目の変化区間である。 $d(x_i, y_j)$ は、 $x_i$ と $y_j$ の時間的な

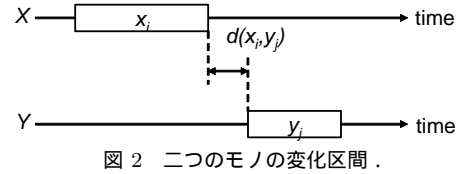


図2 二つのモノの変化区間。

距離である。 $d(x_i, y_j)$ が大きくなれば、Tempが小さくなる考えは正しいだろう。この考えを基に、forgetting factorの概念を利用してTempを下記のように計算する。

$$Temp(X, Y, t) = \sum_{i=1}^N \sum_{j=1}^M \lambda_1^{d(x_i, y_j)} w(x_i) \cdot w(y_j).$$

$\lambda_1$ がforgetting factorで、 $\lambda$ が小さいほど過去の値の影響が小さくなる。また、Nの変化区間  $x_i$  ( $i = 1, \dots, N$ )、および、Mの変化区間  $y_j$  ( $j = 1, \dots, M$ )がXとYに時区間  $t$  に観測されたとする。 $w(x_i)$ は下記のように表される。

$$w(x_i) = \frac{|x_i|}{1/i \sum_{k=1}^i (|x_k|)}.$$

$|x_i|$ は $x_i$ の長さであり、また、過去の変化区間  $x_k$  ( $k = 1, \dots, i-1$ )をあらかじめ保持しているとする。つまり、 $w(x_i)$ は $x_i$ の長さが過去に得られたXの変化区間の長さの平均比べてどの程度大きいかを表す。これにより、普段の利用に比べて短時間の利用(ちょっと触れた程度の動きなど)の重みを抑制できる。以上から、Tempは、時区間  $t$  における、XとYの変化区間の時間的な距離と、履歴から求めた変化区間の重みを考慮した式といえる。

#### [Hist(X, Y)]

Histの計算方法について説明する。あらかじめ  $p$  日分 ( $T_1, \dots, T_p$ )の環境に存在する全てのモノの変化区間を取得しているとする。 $T_p$ が最近の日とする。Histは下記の式を用いて表す。

$$Hist(X, Y) = \frac{h(X, Y) + h(Y, X)}{2}, \text{ where}$$

$$h(X, Y) = \frac{\sum_{i=1}^p \lambda_2^{p-i} Temp(X, Y, T_i)}{\sum_{i=1}^p \sum_{Z \neq X} \lambda_2^{p-i} Temp(X, Z, T_i)}.$$

$h(X, Y)$ は、 $T_1$ から $T_p$ においてYが環境に存在する全てのモノと比べてどの程度Xと一緒に使われたかを表す。ここでもforgetting factor  $\lambda_2$ を用いることで、近い過去を重視している。全てのモノ間のHistは日が変わるごとに計算して保持しておく。

#### [Sem(X, Y)]

Semは語彙の関連性を測るのによく使われるSimpson係数を用いて下記のように表される。

$$Sem(X, Y) = \frac{Hit(X \cap Y)}{\min(Hit(X), Hit(Y))}.$$

ただし、Hit(X)はXの名前を検索エンジンにクエリしたときのヒット数である。

#### 3.1.2 ClusteringとFiltering

ClusteringとFilteringについて概要のみ説明する。Clusteringには、階層的クラスタリングの手法の一つであるWard's法[6]を用いる。このとき、前節で説明したモノ間のDoSをモ

ノ間の距離としてクラスタリングを行う。

*Filtering* では、 $t$  分間における変化区間の合計時間が  $\varepsilon_{ac}$  秒を超えるモノを含まないクラスタをフィルタし、残りのクラスタを出力する。

### 3.2 Web search

上記の手順によってモノの集合からなるクラスタが得られる。そして、クラスタごとにクエリ (複数のサブクエリ) を作成し、クエリに対応する Web ページを取得する。図 1 に示したように、web 検索は主に 3 つの手順からなる。まず、クラスタから複数のサブクエリを作成し (*Query making*)、そのクエリを検索エンジンに送信することで複数の検索結果を得る (*Search*)。クラスタと web ページの類似度を求めることで検索結果をリランクし (*Re-ranking*)、top-1 のページの URL を出力する。*Query making* と *Re-ranking* について下記で説明する。

#### 3.2.1 Query making

この過程において、我々はクラスタにおけるモノの重要度を用いてクラスタをベクトルで表現する。あるモノの重要度は、同じクラスタに属する他のモノとの *DoS* のうち最も大きいものであると定義する。モノの重要度は、クラスタに対応する行動におけるそのモノの貢献度といってもいい。つまりこの場合、モノ間の意味的な関係、ヒストリ、他のモノと近い時刻に使われたかどうか、を考慮した重要度と言える。この手順により、例えば、 $\langle \text{juicer}, 3.0 \rangle$ ,  $\langle \text{cup}, 3.0 \rangle$ ,  $\langle \text{milk}, 2.0 \rangle$ ,  $\langle \text{cup}, 1.0 \rangle$ ,  $\langle \text{sugar}, 0.5 \rangle$  といった重要度付きのモノのリスト (ベクトル) が得られる。ただし、同じ名前のモノが複数含まれるときは、最も大きい重要度をもつモノ以外はクラスタから削除する。この例では、 $\langle \text{cup}, 1.0 \rangle$  が削除される。以上の手順により 4 次元のクエリベクトルが作成される。以下で説明する手法の目的は、このクエリベクトルを用いて行動によく関連するページを見つけることである。*Query making* では下記の 3 つの手法をクラスタに対して順に適用する。

#### [1. ベクトル拡張]

以前の時区間において一緒に使っていたモノの名前がクエリ生成に役立つことは多い。例えば、ある時区間において green-tea と cup を用いてお茶をいれていたとする。そして、その後の時区間において cup を使ってお茶を飲んでいたのである。そのとき、“cup” の名前のみを用いてクエリを作成しても、お茶を飲むという行動に関する web ページを得ることは難しいだろう。しかし、以前の時区間において一緒に使っていた green-tea もクエリ生成に利用することで行動に関するページを取得しやすくなるだろう。まず、注目している時区間におけるクエリベクトル  $Q_i$  の要素数が  $\varepsilon_{hq}$  以下のとき、過去の時区間におけるクエリベクトルのうち、下記に示す  $Sim_h$  が最も大きく、類似度が  $\varepsilon_{hw}$  より大きいクエリベクトル  $Q_j$  を選ぶ。

$$Sim_h(Q_i, Q_j) = \lambda_3^{d(Q_i, Q_j)} \cos(Q_i, Q_j),$$

ただし、 $\cos(Q_i, Q_j)$  は  $Q_i$  と  $Q_j$  のコサイン類似度である。そして、 $\lambda_3^{d(Q_i, Q_j)}$  を乗算した  $Q_j$  を、 $Q_i$  に加える。つまり、過去のクエリベクトルを用いてクエリベクトルを拡張する。実装では、 $\varepsilon_{hq} = 2$ ,  $\varepsilon_{hw} = 0.7$ ,  $\lambda_3 = 0.99$  とした。

#### [2. サブクエリの作成]

1 つのクエリベクトルから複数のサブクエリを作成する。簡単に言うと、クエリベクトルからある程度の数のモノをピックアップし、クエリを作成する。ピックアップして複数のサブクエリ

を作成することで、*Cluster objects* においてクラスタに混入したノイズ (間違っクラスタの要素となっているモノ) を含まないクエリを作成できる。われわれは単純に、クエリベクトルに含まれる全てのモノの組合せからサブクエリを作成する。クエリベクトルに  $i$  のモノが含まれており、所望するクエリ長が  $l$  のとき、 ${}_i C_l$  個のサブクエリが作成される。また、サブクエリの数を制限するため、重要度が  $s$  番目までのモノを含まないクエリは廃棄する。例えば、本小節冒頭で紹介したクエリベクトルの例 (juicer, cup, milk, sugar) からは、 $l = 2$ ,  $s = 2$  とすると、“juicer cup,” “juicer milk,” “juicer sugar,” “cup milk,” “cup sugar,” のサブクエリが作成される。われわれの実装では  $l = 2$ ,  $s = 2$  とした。ComScore [5] によると検索エンジンに送られるクエリ長の平均は、2.54 と言われている (Google, Yahoo!, MSN を含む)。人による作成に近い自然なクエリを作成するため  $l = 2$  とした。

#### [3. クエリ拡張]

object の名前しか含まないクエリはあいまいになることがある。例えば、“cup green-tea” といったクエリから我々が目的とするページ (生活行動に関する追加情報や tips) を得るのは難しいだろう。一方、[8] では、良いクエリを作成するには、トピックに関する語とジャンルに関する語を組み合わせるとよいと言われている。例えばカメラを買いたいときは、“camera” という語と “buying” や “choosing” といった語を組み合わせる (“camera buying” というクエリを作る)。我々も同様に、生活行動に関するページが得られそうなジャンル語を用いてそれぞれのサブクエリを拡張する。われわれの実装では、4 つのジャンル語 (advice, how-to, tips, trivia) から 1 つをランダムに選ぶ。

#### 3.2.2 Re-ranking

*Query making* において作成した複数のサブクエリを検索エンジンに送ると、複数の検索結果 (ランキング) を得ることができる。*Re-ranking* は、クエリベクトルと web ページの類似度を測る尺度を用いて行う。ここで、以降では top-n のページを取得する手続きが多く行われるが、top-n のページは以前の時区間において出力されたページを省いた top-n であることに注意して欲しい。これは同じページを複数回ユーザに提示しないようにするためである。われわれは、下記の 3 つの尺度で類似度を計算するアルゴリズムを用意した。

#### - Term distance algorithm

Web ページ内の近くにクエリ語 (クエリターム) 同士が位置しているページはクエリによくあった文章を含むだろうという観点から、メタサーチの分野においてクエリ語間の距離を考慮した web ページとクエリの類似度指標が提案されている [10]。我々はその指標を web ページとクエリベクトルの比較用に改良する。下記にその指標を示す。ただし、クエリベクトル  $Q = \{ \langle t_1, w_1 \rangle, \dots, \langle t_N, w_N \rangle \}$  である。 $t$  は Porter stemmer [14] によりステミングされた語、 $w$  は語の重要度である。 $W$  は HTML タグが除去され、ステミングされた web ページである。

$$R_d(Q, W) = c_1 N_p(Q, W) + \frac{N_t(Q, W)}{c_3} + \left( c_2 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \min(D_1(t_i, t_j, W), c_2)}{\sum_{k=1}^{N-1} N_p(Q, W) - k} \right) / \frac{c_2}{c_1},$$

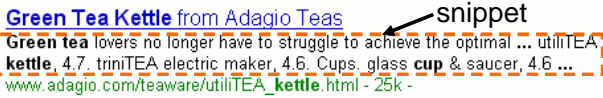


図 3 スニペットの例 .

$$N_p(Q, W) = \sum_{i=1}^N n_p(t_i, W)w_i, \quad N_t(Q, W) = \sum_{i=1}^N n_t(t_i, W)w_i,$$

$$n_p(t, W) = \begin{cases} 1 & (n_t(t, W) > 0) \\ 0 & \text{otherwise,} \end{cases}$$

$n_t(t, W)$  は文書  $W$  に  $t$  が出現する数である .  $D_1(t_i, t_j, W)$  は,  $W$  における語  $t_i, t_j$  間の最小距離 (キャラクタ数) である .  $c_1$  は  $R_d$  の重要度を調整する定数である .  $c_2$  は語間の最大距離を表す整数である .  $c_3$  は語の出現回数の重み付けのための整数である . [10] と同様に  $c_1 = 100, c_2 = 5000, c_3 = 10c_1$  とした . このアルゴリズムでは, それぞれのランキングの top- $(r/\#subqueries)$  をダウンロードし ( $r = 50$  とした .), ダウンロードしたページとの類似度を計算する .

#### - Text analysis algorithm

ダウンロードしたページをさらに詳細に解析することで, 行動に関係する知識やトリアなどを含むページを高いランクにリランクできる . このアルゴリズムは, 語同士の距離に加えて下記を考慮して web ページのスコアを計算し, スコアが最も高いページを出力する .

(1) 事前実験ではショッピングのページや製品紹介ページは ADL に関係することが多いが, 不快に思うユーザもいることが分かった . ページ内のリンクにショッピングサイトへのリンクを多く含むページには負のスコアを与える . われわれは Open directory project [13] の Shopping category に含まれるサイトをショッピングサイトとして用いた .

(2) 事前実験において出力された優良なページを吟味したところ, “housekeeping trivia” といったページ内のサブタイトルや前置きのあとに, モノを用いた ADL に関する知識が記述されているケースが多く見られた . そこで, ジャンル語の後のパラグラフにモノに関するクエリ語が現れているページに正のスコアを与える .

(3) ページの最初に現れるパラグラフはページ内の重要な情報を含む [19] . そのパラグラフにクエリ語が含まれるときは, そのページに正のスコアを与える .

#### - Snippet algorithm

上記の 2 つのアルゴリズムは web ページをダウンロードする必要がある . 一方で近年の PDA や携帯電話といった携帯端末の普及から, われわれはセンサードから得られたセンサデータをそれぞれのユーザの携帯端末上で処理することで, ユーザの端末上にユーザの行動に関連するページを表示することを考えている (携帯端末を用いることで, 屋外環境での web 検索も可能だろう .) . しかし, ページのダウンロードは通信速度や計算能力に制限のある携帯端末にとって現実的ではない .

ここで, ほとんどのサーチエンジンは検索結果にページのスニペット (ページのサマリ) を含んでいる . 図 3 は, “green-tea kettle cup” というクエリから得られた検索結果の一部である . 図のようにスニペットはクエリ語を含むように作成されており, クエリ語同士がページ内において近くに位置しないときは, ‘...’

などのデリミタにより文章が区切られる . 本アルゴリズムはダウンロードした web ページを用いる代わりにスニペットをクエリベクトルとの比較に用いる . クエリベクトル  $Q$  とステミングされたスニペット  $S$  の類似度は下記のように求める .

$$R_s(Q, S) = c_1 N_p(Q, S) + \left( c_2 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \min(D_2(t_i, t_j, S), c_2)}{\sum_{k=1}^{N-1} N_p(Q, S) - k} \right) / \frac{c_2}{c_1},$$

$D_2(t_i, t_j, S)$  はクエリ語間の最小距離 (キャラクタ数) である . ただし, クエリ語間にデリミタがあるときは  $c_2$  を返す . また, スニペットは長さが短いため,  $N_t$  は用いない . このアルゴリズムでは, それぞれのランキングの top- $(r/\#subqueries)$  のスニペットとクエリベクトルの類似度を  $R_s$  を用いて計算し, web ページをリランクする ( $r = 200$  とした .) .

## 4. 関連研究

### 4.1 ページの適合と割り当て

本論文は日常生活に関する web ページを取得する方法について着目している . ここでは, PC 用に作成された web ページを大きなディスプレイをもたない機器用に変換する研究や, サービス割り当てに関する研究について紹介する . 本研究では, 身の回りにあるインターネットにつながった様々なデバイスに web ページが出力される . [3] は, web ページ中のテキストをサマライズすることにより画面の小さい PDA などのデバイスに適合させている . [4] は, web ページを複数のブロック (関連する情報の塊) に分割し, そのブロックごとに小さなディスプレイに出力するための手法を提案している . また, [11] では, ブロック分割したページを小さなディスプレイに順次提示するアプリケーションを実現している . [12] では, 目の不自由なユーザのために, web ページの読み上げアプリケーションを実現している . これらにより, 1. で述べた小型のディスプレイを持つ電子レンジや冷蔵庫といった家電や, 音楽プレイヤーなどの音声出力のできるデバイスに web ページを出力できるだろう .

[1] では, 無線通信の周波数を用いた, 無線機器 (例えば無線 LAN を備えたラップトップ PC) の位置検出を実現している . 一方で, ユビキタスコンピューティング研究の分野では, ルールを用いて身の回りにあるさまざまな機器にコンテンツを割り当てる (再生する, 表示する) ことが多い [2] . これらの研究を利用することで, 行動が行われている位置に応じて web ページを身の回りの機器に割り当てることが可能になると考える . 例えば, “音楽プレイヤーの近くで行われている行動に関する web ページは, その web ページ内の文章を音声に変換してそのプレイヤーで再生する .” などのルールが記述できるだろう .

### 4.2 クエリフリーサーチとコンテキストサーチ

我々の知る限り, 日常生活において利用されるモノを用いた web 検索は他に存在しない . ここでは, 我々の研究とよく関連するクエリフリーサーチとコンテキストサーチの研究について紹介する . まず [7] は, TV ニュースのテロップからクエリを自動的に抽出し, クエリを検索エンジンに送ることで, そのニュースに関する web ページを取得する . [9] では, 書いているドキュメントや閲覧している web ページ中の語を用いて, ユーザが入力しているクエリを強化する手法を提案している . さらに, 複数のサブクエリを作成し, それらから得た複数のランキ

ングを Markov chain を用いてリランクしている .

Web ページではなく、ローカル PC に保存されているドキュメントを提示する研究も存在する [15] は、ユーザが Emacs を使って書いているドキュメントに関する他のドキュメントをローカル PC から検索し提示する . 検索クエリは、ユーザがドキュメントを作成する際に入力した語から作成される . [16] は、ウェアラブルコンピュータのためのセンサデータを用いたクエリフリーサーチを実現している . 例えば、GPS センサにより得たユーザの現在地に関するドキュメントをローカル PC から検索し、Head Mounted Display に提示する .

### 4.3 状況依存型サービス

多くの状況依存型サービスは、ルールベースシステムにより実現される . つまり、ある条件を満たせば、あらかじめ用意したサービスを提供するといったルールを備えることで、コンテキストに合わせたサービスを提供するシステムである . ユビキタスコンピューティング研究では、条件にセンサの出力信号やセンサデータを用いた ADL の推定結果が用いられることが多い . 例えば [18] では、ECA (Event, Condition, Action) ルールを用いたシステムを実現しており、人が部屋に入ったことがセンサにより検知されれば部屋の照明や空調を自動的に調整するアプリケーションなどを紹介している . 我々が提案する ADL のための web 検索も、センサを使った状況依存型サービスの一つとっていいだろう . しかし、我々の提案はコンテキスト (ADL) を推定せずに、web ページを提供する点でルールベースの状況依存型システムとは異なる (多くの場合、ADL の推定にはその環境における教師信号を必要とするため、ユーザにかかる負担が大きい) . さらに、我々の提案はコンテキスト (ADL) ごとの人手によって作成されるルールを必要としない点でも異なる .

## 5. む す び

本論文では、ユビキタス・センサネットワーク環境における、ユーザの日常行動のための web 検索手法を提案した . 提案手法は、主に利用されたモノをクラスタリングする手法と使われたモノの名前から web ページを検索する手法からなる . クラスタリング手法では、モノ同士と一緒に使われたかどうか、過去においてモノ同士と一緒に使われていたかどうか、モノ同士の意味的な関連性からクラスタリングを行う . Web 検索手法では、クラスタリングされたモノの名前からクエリを作成し、検索エンジンから得られた検索結果をリランクする . リランク手法は目的に合わせて 3 種類を用意した . 今後はクラスタリング手法も併せて実環境で得られたセンサデータを用いて評価を行う予定である .

## 文 献

- [1] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *Proc. INFOCOM 2000*, pp. 775–784, 2000.
- [2] M. Beigl, A. Schmidt, M. Lauff, and H. Gellersen, "The ubi-compbrowser," *Proc. 4th ERCIM Workshop on User Interfaces for All*, 1998.
- [3] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, "Seeing the whole in parts: text summarization for web browsing on handheld devices," *Proc. WWW-10*, 2001.
- [4] Y. Chen, W.Y. Ma, and H.J. Zhang, "Detecting webpage structure for adaptive viewing on small form factor devices," *Proc. WWW 2003*, pp. 225–233, 2003.
- [5] comScore, Inc., <http://www.comscore.com/>.
- [6] J.F. Hair, R.E. Andersen, R.L. Tatham and W.C. Black, "Multivariate data analysis," 4th ed. Prentice-Hall, Englewood Cliffs, N.J.
- [7] M. Henzinger, B.W. Chang, B. Milch, and S. Brin, "Query-free news search," *Proc. WWW 2003*, pp. 1–10, 2003.
- [8] R. Kraft and R. Stata, "Finding buying guides with a web carnivore," *Proc. the 1st Latin American Web Congress (LA-WEB)*, pp. 84–92, 2003.
- [9] R. Kraft, C.C. Chang, F. Maghoul, and R. Kumar, "Searching with context," *Proc. WWW 2006*, pp. 477–486, 2006.
- [10] S. Lawrence and C.L. Giles, "Inquirus, the NECI meta search engine," *Proc. WWW-7*, pp. 95–105, 1998.
- [11] T. Maekawa, T. Hara, and S. Nishio, "Image classification for mobile web browsing," *Proc. WWW 2006*, pp. 43–52, 2006.
- [12] J. Mahmud, Y. Borodin and I.V. Ramakrishnan, "CSurf: a context-driven non-visual web-browser," *Proc. WWW 2007*, pp. 8–12, 2007.
- [13] Open Directory Project, <http://www.dmoz.org>.
- [14] M.F. Porter, "An algorithm for suffix stripping," *Program*, 4, pp. 130–137, 1980.
- [15] B. Rhodes and T. Starner, "The remembrance agent: a continuously running information retrieval system," *Proc. Int'l Conf. on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM96)*, pp. 486–495, 1996.
- [16] B. Rhodes, "The wearable remembrance agent: a system for augmented memory," *Personal Technologies: Special Issue on Wearable Computing 1*, 218–224, 1997.
- [17] J. Sohn, N.S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, 6, pp. 1–3, 1999.
- [18] T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, A. Kashitani, and S. Nishio, "Ubiquitous chip: a rule-based I/O control device for ubiquitous computing," *Proc. Pervasive 2004*, pp. 238–253, 2004.
- [19] A. Tombros and M. Sanderson, "Advantages of query biased summaries in information retrieval," *Proc. SIGIR 1998*, pp. 2–10, 1998.