4.1 メタデータに基づく情報検索

WWW 情報検索において、検索者は属性情報(時間や場所、状況、嗜好などの情報)をキーワードによって入力している。検索者の望む情報を効率よく検索するには、検索者の変化する属性を反映する情報検索が望まれる。そのため、本研究では記述内容の分類、属性情報を活用した検索手法の2点に着目し機能要件を整理する。そして、機能要件をもとに検索者の発するクエリおよび検索対象に属性記述モデルに即したメタデータを付加し、それらを照合する情報検索手法を提案する。また、提案に基づき検索エンジンを実装し、評価実験を行った結果、全文検索よりも高いヒット率および短い目的到達時間での情報検索が可能となった。

4.1.1 はじめに

現在、有線接続による固定端末でのインターネットの利用以外に、無線 LAN、携帯電話などの無線接続による移動端末でのインターネット接続も可能となっている。これにより、移動先の予定時間における天気情報の取得や、現在地周辺で検索者の嗜好に合致した飲食店の検索といった検索者の状況に依存した情報検索が多くなった。

検索者は日々生活を送る上で、時間や場所、嗜好、目的といった属性が変化する。検索者の得たい情報はこのような属性の変化に伴い異なってくる。WWW コンテンツの検索手法としてはコンテンツと検索クエリのテキストのパターンマッチングによる全文検索が主流となっている。全文検索では検索者が自らの属性の変化を判断し、検索キーワードを入力している。

一方、検索対象となる WWW コンテンツも本文中には時間や場所、目的などの属性が含まれている。しかしこれらの情報は属性を明示的に示しておらず、属性は単なるテキスト記述となっている。したがって計算機の機械処理による属性の抽出が困難である。

検索対象となるコンテンツに属性情報を明示的に付加して機械処理を容易とする手段としてメタデータがある。

WWW コンテンツに対するメタデータ付加については、W3C(World Wide Web Consortium)[1]の Semantic Web[2,3]において多くの研究がされており，メタデータの書式として、RDF(Resource Description Framework)[4]が規定

されている。

全文検索において検索者が自らの属性の変化を判断し検索キーワードを入力しても、意図した属性が検索結果に反映されないことがある。またメタデータを利用した既存の情報検索はコンテンツ側にのみメタデータを付加しているものがほとんどで、検索者のクエリについては項目を分けてフォーム入力する必要がある。検索毎に複数の項目に大量のキーワードを入力することは検索者にとって大きな負荷となる。これはユーザインターフェイス能力が限られた携帯情報端末等を用いて情報検索を行う場合はさらに深刻となる。これらの問題点に着目しつつ、天気予報やグルメ情報など生活に密着した情報を効率よく検索することと、スコアリングの困難さのバランスを考慮して、属性情報を人の行動の要素となる Time, Position, Occasion の 3 属性に分類する。

本研究では、TPO によって分類されたメタデータを検索対象および検索者のクエリの元となる要素に付加させ、それぞれのメタデータを属性毎にマッチングしスコアリングする検索手法を提案する。

4.1.2 WWW 情報検索の要求事項

メタデータは「データのデータ」と定義され情報の内容を定義するとともに情報を構造化することを目的としたものである[5]。したがって検索者の属性情報に適応する検索を実現するには検索対象に属性情報が記述されたメタデータを検索することが有効である。メタデータ検索を実現するためには、検索対象のコンテンツの意味を示すメタデータ、検索者の検索要求を記述したメタデータ、およびそれらのメタデータのマッチング機能、マッチング結果の評価機能が必要となる。

ネットワーク上のコンテンツに対する属性情報の付加はメタデータを用いることが最も適している。検索者のクエリも属性に分類し、生成しなくてはならない。検索者は Web 上の検索フォームにおいて属性毎に独立してクエリの元となるパラメータを入力する必要があり、これは 1 つの検索にかかる労力が増加する。普段変化の少ない検索者の属性情報をメタデータによって保存しておき検索クエリ作成を補助することによって、検索労力が削減できる。

4.1.3 TPO 記述モデルを利用したメタデータ検索の提案

TPO に基づいた属性情報をメタデータとして、コンテンツのメタデータと検索者が発するクエリの双方をメタデータによって記述し、メタデータをマッチングすることによって情報検索する手法を提案する。提案モデルの概略は図 1 のようになる。コンテンツの TPO 属性を記述したメタデータをコンテンツメタデータ、検索者のクエリとなるメタデータをユーザメタデータと定義する。メタデータ検索エンジンはコンテンツメタデータとユーザメタデータを比較(マッチングおよびスコアリング)して、TPO 属性がマッチしたものを検索結果として検索者側に提示する。

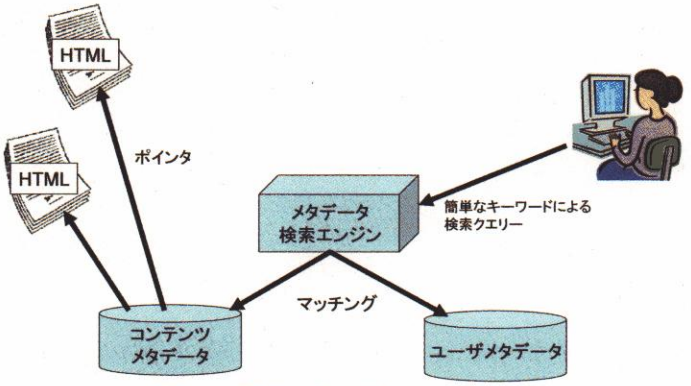

図 1: 提案する検索モデル

4.1.3.1 メタデータの分類

検索対象となる Web 上の文書類のメタデータは基本的に静的であり変化しないと仮定する。 それに対して、検索者のメタデータは TPO の変化に伴い値が動的に変化する。 このような動的なメタデータを取り扱うためには、まず動的なメタデータが取り得る属性と値を規定する必要がある。

検索者の TPO 属性の変化に影響を受けるメタデータの値は、メタデータの種類に依存して更新頻度や保持すべき情報が異なる。例えば、時間に関するメタデータは、検索者のスケジュールを反映させたメタデータを定義した場合、スケジュールが追加および変更される度にメタデータを更新する必要がある。それに対して現在時刻はメタデータとして保存する必要性はなく、リクエストの際に端末の時刻を抽出すればよい。

コンテンツメタデータはユーザメタデータに比べ属性の変化が緩やかであ

るため、基本的にユーザメタデータが取り得る値を包括していればよい。

本研究で取り扱う属性情報は、キーワードによって記述できない属性情報と比較的変化の少ない検索者の属性情報がある。キーワードによって記述できない属性情報は、時間や距離といった数値情報となる。店舗情報の開店時間を例に挙げると、Web コンテンツ中には開店時刻および閉店時刻のみが記述されている。検索者が現在時刻において開店している店舗を検索しようとしても従来の全文検索等による文字パターンマッチングを用いた検索手法では不可能である。距離についても同様のことが予測できる。キーワードによって記述できない情報は基本的に数値によって記述されている情報が主となる。比較的変化の少ない検索者の属性情報は、文字パターンマッチングによる検索は可能であり、基本的にテキストによって記述される。

TPO の各属性は意味情報として次元が直交していると仮定し、メタデータを TPO に分けて定義する。Time 属性情報については現在時間やスケジュール情報を規定する。Position 属性情報については現在地情報や移動範囲、また電話番号等も Position 属性情報として扱うことが可能である。Occasion 属性情報については、目的や手段、検索対象のジャンルなど嗜好情報を含んだものが主になる。

テキスト記述によるコンテンツメタデータはスコアリングの指標が必要となるので、重要度を設定する。メタデータを分類した結果は図 2 のようになる。メタデータはまずユーザメタデータかコンテンツメタデータかに分類される。次に TPO によって分類され、属性の記述形式に分類される。分類された属性は基本的に一つの値を持つ。テキスト記述によるユーザメタデータについてはスコアリングの指標を考慮して重要度が設定される。ユーザメタデータの現在時刻を例に挙げると、所有者は検索者、TPO 分類は T、属性の性質は数値、属性名は現在時間、値は 15:00 といった情報になる。

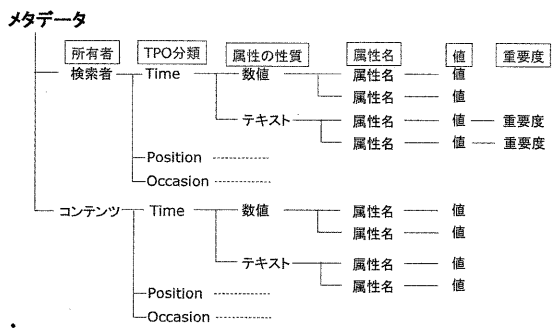

図 2: メタデータの分類

4.1.3.2 メタデータを用いた情報検索手法

検索の作業はマッチングとスコアリングに分けられる。ユーザメタデータが示す属性に該当するコンテンツメタデータをコンテンツメタデータ群から抽出する作業がマッチングで、得られたマッチング結果を検索者にとって重要で的確な情報ほど上位に表示するための指標を算出する作業がスコアリングである。

- マッチング

TPO 各要素で独立のマッチング作業を行い、それらの結果の論理和集合をマッチング結果とする。数値をもつメタデータはその数値に適した演算によってマッチングする。テキスト情報をもつメタデータについては同属性のコンテンツメタデータ中にユーザメタデータに記述されているテキストが含まれている場合はマッチングしたと判断する。基本的なマッチング方法は全文検索と同様だが、検索対象が同属性内に絞られることにより検索の効率と精度の向上が期待できる。

- スコアリング

マッチング結果は TPO に沿って記述内容の次元が直交しているので、TPO の各要素で独立してスコアリングする必要がある。各々のスコアリング結果を同じ場で演算するために正規化し、得られた 3 つの指標に重みづけをして最終的なスコアを得る。スコアはユーザメタデータによって動的に変化するため全文検索エンジンのようなコンテンツへの静的なランク付けが困難である。そのため各スコアリング式の結果を平均値との差分をとって擬似的に正規化する。

4.1.4 検索エンジンの設計

ここでは TPO 記述モデルを利用したメタデータによる検索エンジンの設計について述べる。設計全体図を図 3 に示す。検索者はまず検索システムにユーザ登録し、ユーザメタデータを保存するユーザディレクトリを作成する。検索者は作成したユーザディレクトリにユーザメタデータをあらかじめ保存しておく。検索者の属性が変化した時はユーザメタデータの更新を検索システムを介して行う。提案手法の検索エンジンは複数のユーザメタデータを保存する。またコンテンツメタデータについてはコンテンツ作成者があらかじめメタデータを付与していることを想定している。検索システム側はロボットによってコン

テンツのメタデータを収集し、検索エンジンの作業効率を考慮して、インデキシングエンジンによりメタデータを 2 次元テーブルに保存しておく。

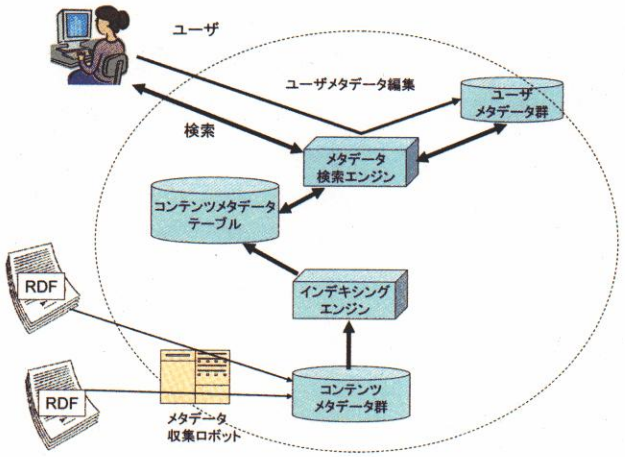

図 3: メタデータ検索エンジン

検索者は目的のコンテンツを検索したい時に簡単なキーワードを入力して検索エンジンにクエリを送信する。検索エンジンは検索者から受け取ったクエリをユーザメタデータに変換して、あらかじめ検索者のユーザディレクトリに保存しているユーザメタデータと合わせて検索用のユーザメタデータを揃える。検索エンジンは揃えたユーザメタデータとコンテンツメタデータをマッチングおよびスコアリングし、検索結果を検索者に返す。

4.1.4.1 マッチング部

TPO および記述形式の分類に従ってマッチング作業を行い、それらの条件を全て満たすものをマッチング結果とする。マッチング判定式を以下に示す。

$$A = \coprod_{i=T,P,O} \prod_{j=1}^{n_i} fm_{ij}(M_{fm_{ij}1}, M_{fm_{ij}2}, \cdots, M_{fm_{ij}k})$$

$$fm_{ij} = \begin{cases} 1(matching) \\ 0(not-matching) \end{cases}$$

$M_{fm_{ij}k}$: 引数となるメタデータ

分類毎のマッチングはあらかじめ定義される関数によって行われる。引数は関数によって数が異なる。

検索者が現在時刻において店舗情報を検索した場合のマッチングを例に挙げると、現在時刻と開店時間のマッチングにおいて引数は、ユーザメタデータの現在時刻、コンテンツメタデータの開店時刻、閉店時刻の 3 つとなり、ユーザメタデータの現在時刻がコンテンツメタデータの開店時刻と閉店時刻の間にある場合はマッチングしたと判定される。

コンテンツのマッチング結果は、すべてのマッチング式の結果をかけあわせた結果 A で、1 か 0 で出力される。

4.1.4.2 スコアリング部

マッチング結果が $A=1$ となったコンテンツのうち検索者の属性を反映している度合いが高いものを、結果表示の上位に表示するために各コンテンツのスコアを評価する。スコアリングの評価式を以下に示す。

$$S=\sum_{i=T,P,O}\sum_{j=1}^{n_i}\frac{k_i}{n_i}\bullet\frac{fs_{ij}-\overline{fs_{ij}}}{\overline{fs_{ij}}}$$

$$k_T+k_p+k_o=1$$

$fs_{ij}\left(M_{fs_{ij}1},M_{fs_{ij}2},\cdots,M_{fs_{ij}k}\right)$：スコアリング式

k_T,k_P,k_O：TPO の重み係数

各スコアリング関数毎に平均値との差分を平均値で割ることによって擬似的な正規化を行っている。また TPO 各属性毎のスコアの総和を、TPO 各属性に存在し得る関数の総数 n_i で割ることによって擬似的な正規化を行っている。

スコアリング関数の出力は検索者の属性を反映しているものほど値が高くなるように式を生成する必要がある。検索者の現在位置と検索対象の存在位置との距離をスコアリングする場合は距離が近ければ近いほどスコアが高くなるような式を設定すればよい。

4.1.5 検索エンジンの評価

実装した内容はメタデータ生成部、インデックス生成部、個人メタデータの

作成・保持部、検索エンジンであり、すべて同一 PC 上で実装した。システム全体部はデータベースとの連携処理が容易にできることと、クライアント側になるべく制約を設けないようにサーバサイドスクリプティングを選択したことから、PHP を用いた。

実験対象の Web ページはコンテンツに含まれる属性の多彩さと記述の規則性があることを考慮して Yahoo!のグルメ情報(大阪府内 2233 件)とした。
スクリプト処理によって抽出したメタデータは、Time として、開店時間、閉店時間、定休日、Position として、緯度、経度、Occasion として、店名、平均予算、ジャンル、目的、メニュー、扱えるクレジットカード、コメントがある。また TPO に基づいて生成されたメタデータは実際に図 4 のように記述される。

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ut="http://hoge.aist-nara.ac.jp/classes/">
  <rdf:Description
    rdf:about="http://hoge.aist-nara.ac.jp/gourmet/hoge.html">
    <ut:Time>
      <rdf:Description>
        <ut:open>11:00</ut:open>
        <ut:close>22:00</ut:close>
      </rdf:Description>
    </ut:Time>
    <ut:Position>
      <rdf:Description>
        <ut:latitude>34.72</ut:latitude>
        <ut:longitude>135.73</ut:longitude>
      </rdf:Description>
    </ut:Position>
    <ut:Occasion>
      <rdf:Description>
        <ut:name>ismail cafe</ut:name>
        <ut:budget>1000</ut:budget>
        <ut:genre>cafe</ut:genre>
        <ut:purpose>date</ut:purpose>
        <ut:menu>latte, mocha</ut:menu>
      </rdf:Description>
    </ut:Occasion>
  </rdf:Description>
</rdf:RDF>
```

図 4: 生成されたメタデータ

4.1.5.1 実験方法

本研究で提案した検索エンジンが 4.4.2 の要求事項を満たしていることを確認する。ここでは、主に属性情報の分類と属性情報を活用した検索システムの有効性について述べる。実験は実装環境と同様に PHP スクリプトが動作可能な WWW サーバをインストールしたデスクトップマシンを利用する。 クライ

アントについては検索者が所有しているノートPCを利用した。

提案した検索エンジンでは検索対象となるネットワーク上のWebコンテンツにメタデータを用いて属性情報を付加した。これによって従来の全文検索型サーチエンジンより検索者の属性に適応した情報検索が実現したか否かを評価する。

10人の被験者にシナリオを提示してグルメ情報の検索をしてもらい、既存のYahooグルメによるカテゴリ検索、全文型検索エンジン(日本HP社のMitakeSearch)による検索、および提案検索エンジンによる検索の3種類の検索結果の比較実験を行った。提示するシナリオは3つで、内容はシナリオAを「大阪駅の周辺で昼食をとる飲食店を検索する」、シナリオBを「心斎橋で22時に飲みに行く」そしてシナリオCを「午前4時になんばでラーメンを食べる」とする。

提案検索エンジンでの検索では数値情報などのキーワードによって入力できない属性情報は計算機による自動取得もしくはWebのフォームによってあらかじめ登録しておく。テキスト情報については全文検索と同様に、用意されたテキストボックスにキーワードを入力する事によって検索する。

評価点は以下の点で主に投稿フォームによるアンケートにより評価を集計した。

4.1.5.2 上位検索ヒット数

検索者の属性に適応した情報が実現したか否かを評価する。提案した検索エンジンは検索者の属性情報を考慮したスコアリングによるため、検索結果の上位部は全文型検索エンジンに比べて、より検索者の意図に即したものが表示されるはずである。

検索者に提示したシナリオに従って5回検索をしてもらい検索結果の上位10個に表示される検索結果のうち、いくつがシナリオに加えて検索者の意図した目的に適合したかを検索者による判断で計上する。

シナリオAの上位ヒット数の分布を図5、シナリオBのものを図6、シナリオCのものを図7に示す。一人につき一つのシナリオを5回検索した上位ヒット数の平均値を1つの要素として分布を示している。すべての結果において提案した検索エンジンの上位ヒット数の方が全文検索のものより高い値の分布が得られた。シナリオがAからCに変化するにつれて分布の差が顕著に現れた。

シナリオCはシナリオAに比べて時間等の条件が厳しくなっていて比較的検索が困難である。上位ヒット数(個)の平均値は、シナリオA, B, Cそれぞれにおいて、全文検索では、2.0, 3.6, 0.6 となり、提案した検索エンジンでは、2.8, 5.4, 1.9 となり、すべてのシナリオにおいて提案した検索エンジンの方が全文型検索のものに比べて高い結果となっている。

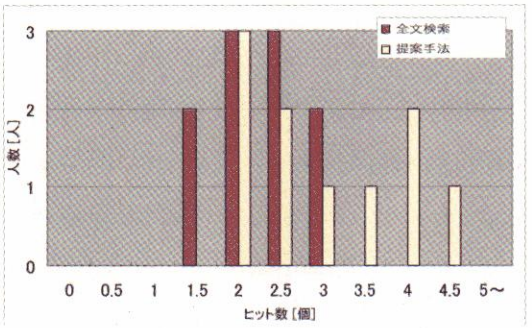

図 5: シナリオAでの上位ヒット数の分布

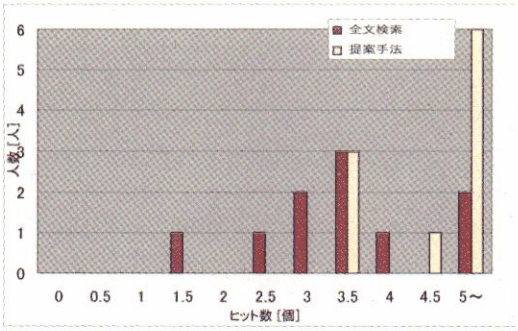

図 6: シナリオBでの上位ヒット数の分布

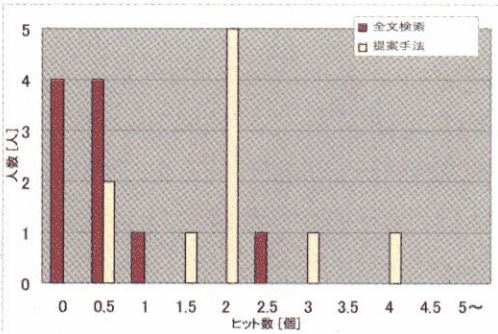

図 7: シナリオCでの上位ヒット数の分布

4.1.5.3 目的到達時間

目的の Web ページにたどり着くまでの時間を測定する。検索ページのトップ画面が表示されてから検索者が提示したシナリオに従って飲食店を検索し、検索者が実際に行こうと決断する店が発見されるまでの速度を目的到達時間とする。この速度は普段変化の少ない検索者の属性情報をメタデータによって保存しておき、検索クエリ作成を補助することによって検索労力がいかに削減できたかを示す指標となる。検索者にストップウォッチを持たせて検索者各自に目的到達速度を測定してもらった。

目的到達時間(秒)の平均値は、シナリオ A, B, C それぞれにおいて、全文検索では、45.1, 47.3, 63.2 となり、提案した検索エンジンでは、48.8, 34.0, 33.4 となった。シナリオ A においては両者において大きな差はないが、シナリオ B、シナリオ C と条件が厳しくなるにつれて目的到達時間の差が大きくなり、提案した検索エンジンでは検索者の属性の変化に対応した情報検索を実現していると判断できる。

4.1.6 おわりに

本提案では TPO 各属性毎に存在するメタデータに必要な数のスコアリング評価式を設定している。各スコアリング式から算出されるスコアはメタデータの適合度に比例して大きくなる。しかし、提供されるスコアリング式には検索者の意図を反映しないものも存在する。本提案では総スコアは各々のスコアを擬似的に正規化して和をとったものである。したがって、検索者の意図を反映しないスコアが大半を占めた場合、検索者の意図を忠実に反映している一部のスコアが埋没してしまう可能性がある。よって検索者の意図に沿うスコアリング関数のみを動的に選択する手法を取り入れるような機構が必要となる。

また、本提案の検索手法は検索毎にスコアリングするため、検索対象がさらに拡大した場合に検索速度が著しく低下するおそれがあるため、複数のマッチング関数の効率的な順序付けについて考慮する必要がある。

今回の評価は検索者によるアンケートであったが、これは複数の実験を繰り返す上で検索者にとって多大な負荷となるため、機械処理による検索エンジンの評価手法についても、今後調査する必要がある。

参考文献

[1] “The World Wide Web Consortium”, http://www.w3c.org/

[2] 萩野達也ほか, “セマンティック web とは”, 情報処理, Vol.43, No.7, pp.709-717, 2002年7月.

[3] “semanticweb.org - together towards a web of knowledge”, http://www.semanticweb.org/

[4] “Resource Description Framework (RDF) / W3C Semantic Web Activity”, http://www.w3c.org/RDF/

[5] 杉本茂雄, “メタデータについて - Dublin Core を中心として -", 情報の科学と技術, Vol.49, No.1, pp.3-10, 1999年1月.