Web 閲覧行動に応じたマルチファセットの動的生成と比較ページの検索

川野　悠† 大島　裕明† 田中　克己†

† 京都大学大学院情報学研究科社会情報学専攻　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †{kawano,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし　本研究では閲覧中の Web ページに複数のファセットを動的に生成する手法を提案する．本研究におけるファセットとは閲覧中の Web ページと比較ページを比べるための観点である．閲覧中の Web ページを表す複数の特徴語に対して，上位語と同位語を発見することでそれぞれファセットを動的に生成する．さらに生成されたファセットから興味のあるものを選択することで，ユーザは選択されたファセットに沿った比較ページの検索を行うことができる．本研究ではユーザの Web 閲覧履歴を利用することで，個々のユーザにとってより重要なファセットの生成を試みた．

キーワード　ファセット生成，Web 閲覧履歴，比較ページ検索

1. はじめに

近年のインターネットの普及により Web ページの数は爆発的に増加し，Web から様々な情報を得ることが可能となった．ユーザは目的の情報を得るためにブラウザを用いて Web ページの閲覧を行うことができる．ユーザの求めている情報はひとつの Web ページに書かれているとは限らず，様々な Web ページを閲覧することで得られるという場合も多い．このようにユーザが目的の情報を求めて複数の Web ページを閲覧する場合，ページ間を遷移する手段として主に，ページ内のリンクを用いる方法と検索エンジンを用いる方法の 2 つが挙げられる．しかしこれらの方法によって，ユーザは必ずしも目的の情報が得られる Web ページに辿り着くことができるとは限らない．例えば，京都にあまり詳しくない人が「八坂神社」で行われる「行事」に関する情報を網羅的に知りたい場合を想定し，現在「八坂神社」の「祇園祭」に関するページを閲覧しているとする．ページ内のリンクはそのページの作者によって作成された静的なものであり，ユーザが求める情報が書かれたページへのリンクが存在するとは限らない．

また検索エンジンを用いることで,「八坂神社　行事」に関するページがリスト形式で提示された検索結果を得ることができるが，その結果には「八坂神社」で行われる「行事」に関するページが混在しており，ユーザは検索結果で提示されたページをひとつずつ閲覧してページの適合性（「八坂神社」で行われる「行事」に関するページかどうか）や適合していれば行事の内容などを確認しながら情報を収集することになる．しかし，京都にあまり詳しくないユーザにとってそのようなことを正しく判断できる保証はなく，かかる負担は大きいと考えられる．この場合，閲覧ページに「八坂神社」で行われる「行事」の一覧が提示され，その中のある行事，例えば「花灯路」を選択すると「八坂神社」と「花灯路」に関するページに遷移することができれば,「八坂神社」の「行事」に関する情報をその行事ごとに網羅的に収集できる．

しかし,「八坂神社」の「祇園祭」に関するページを閲覧しているユーザが常に「行事」に注目しているとは限らない．もし「祇園祭」を見ることができる「観光名所」の情報を網羅的に知りたい場合は，そのような「観光名所」の一覧を閲覧ページに提示する方が望ましい．このように Web ページは一般的に複数のテーマを含んでおり，閲覧中のページと比較されるページへのナビゲーションを考える際にはこのことを考慮する必要がある．

そこで本稿では，閲覧中のページに複数のファセットを動的に生成する手法を提案する．ファセットとは物事を見る側面のことで，あるオブジェクト集合を分類する際の観点として用いられることが多い．例えば車の分類を考えたとき,「製造メーカー」や「製造年」,「色」など様々な観点から分類することができ，T 社の 2000 年製造の赤い車は，製造メーカーファセットでは「T 社」，製造年ファセットでは「2000 年」，色ファセットでは「赤」にそれぞれ分類される．ファセットは「製造メーカー」のような観点を表すファセット名と「T 社」や「N 社」など実際の分類のラベルを表すファセット値から構成される．本研究では全 Web ページを分類の対象とした際の閲覧ページが分類されるようなファセットを発見することで，そのページと比較されるようなページの検索を行う．例えば「八坂神社」の「祇園祭」に関するページは「行事」というファセット名の「祇園祭」というファセット値に分類できるが,「行事」ファセットの別のファセット値である「花灯路」や「節分祭」を提示することで比較ページへの遷移を支援することが期待できる．また複数のファセットを考慮することで,「八坂神社」と「祇園祭」という複数のテーマを含んだ Web ページに対するナビゲーションを可能となる．

本研究では以下のような手順でファセットを動的に生成する．

（1） 閲覧ページの特徴を表す複数の語を抽出する
（2） 抽出したそれぞれの語に対し，同位語と上位語の発見を行うことでファセットを生成する
（3） 生成したファセット名の下位語発見を行うことでファセッ

トの検証を行う

本手法では閲覧ページに複数のファセットが提示されるが，数多くのファセットの中から自分の興味のあるファセットを探し出すことはユーザにとって負担のかかる行為である．そこで，Web ページの閲覧履歴を用いることでユーザが興味を持っているファセットを推定し，そのようなファセットから順番に提示することを考える．例えば，現在「八坂神社」の「祇園祭」に関するページを閲覧していても過去に「金閣寺」や「銀閣寺」に関するページを閲覧していれば「観光名所」というファセットに興味があると予想できるが,「葵祭」や「時代祭」に関するページであれば「祭り」というファセットに興味があると予想できる．ファセットをユーザが興味を持っている度合いが高いと思われる順にランキングし提示することで，より個々のユーザに適したナビゲーションを支援する．

本稿の構成は以下のとおりである．2 節では関連研究について言及する．3 節では閲覧ページに提示するファセットについて述べ，比較ページ検索に効果的なファセットの定義を与える．4 節では Web 閲覧履歴を用いたファセットの生成手法について述べる．5 節では実験と評価，6 節では生成したファセットを基に比較ページを検索する手法について述べる．最後に 7 節でまとめを述べる．

2. 関 連 研 究

2.1 閲覧中の Web ページの提示

吉田ら [1] は Web 閲覧履歴を用いることでユーザが興味を抱いている話題を推定し，ユーザとの対話操作を経て関連ページを検索するシステムを提案した．Web 閲覧履歴を用いてユーザが興味を抱いている話題を推定する点は似ているが，本研究はファセットを提示することで Web ページを比較する基準ごとに分類している点で異なる．

2.2 ファセットを用いた分類

ファセット検索は情報検索の様々な分野で適用されている．[2] [3] [4] [5] [6] 画像検索の分野では，Yee ら [7] は与えられた画像をそれぞれファセットに割り当てることで，複数のファセットから画像を検索できるシステムを考案した．このシステムはインターフェースとして Flamenco [8] を採用している．本手法がファセットをクエリに応じて動的に抽出するのに対し，このシステムは Web 画像検索ではなくドメインが限定された画像集合内での検索であり，あらかじめ決められた静的なファセットに分類するという点で本手法とは異なる．

2.3 特定の関係にある語の抽出

分類構造を構築するにあたって，上位関係や下位関係などの，特定の関係にある語を抽出する技術は非常に重要である．本稿では，ある語 A と特定の関係にある語を A の関連語と呼ぶことにする．WordNet [9] は英語の概念辞書であり，語の簡単な定義や語同士の関係が記されている．しかし WordNet は人工的な辞書なので，全ての語を網羅することは不可能である．大規模なテキストコーパスやデータセットを利用して関連語を抽出する手法にも様々なものがある．Hearst ら [10] は“ such as ”のような言語パターンに着目して，上位語や下位語を抽出する手法を提案した．Ghahramani ら [11] はベイズ推定を用いて，同位語を共起テーブルのような大規模データから発見する手法を提案した．ここで同位語とは共通の上位語を持つような語である．このアルゴリズムは単純かつ高速であるが，EachMovie や Grolier encyclopedia のような大量のデータセットを必要とする．新里ら [12] は HTML 文書から同位語を発見する手法を提案した．大島ら [13] [14] はある語の前後に接続する 2 種類の構文パターンを用いて関連語を抽出する手法を提案した．

3. 閲覧ページに提示するファセット

3.1 ファセットの種類

前節で述べたように，本研究で扱うファセットは 1 つのファセット名と複数のファセット値から構成される．ここで Web ページの遷移を支援するためのファセットとしてどのような種類のものが適切であるかについて考える必要がある．例として，以下のようなファセットが挙げられる．

- 真か偽を問うようなファセット
- 序列に関するファセット
- 語同士の概念階層を表すファセット

真か偽を問うようなファセットとは例えば，ファセット名が「祇園祭かどうか」といったもので，対応するファセット値は「真」と「偽」のみである．序列に関するファセットとはファセット名が「アルファベット順」や「50 音順」となり，ファセット値としては「A」から「Z」または「あ」から「ん」までを順番に羅列したものなどが考えられる．このタイプのファセットは他のファセットとは異なり，ファセット値が何らかの順序に従って並べられていることによって初めて効果が発揮される．例えば本の索引はあるページに書かれている項目へのナビゲーションだと考えられるが，もし索引が「50 音順」ではなくランダムに並んでいたとしたら，読者は探したい項目の頭文字を索引から注意して探し出さなければならず，索引のメリットが半減してしまう．語同士の概念階層を表すファセットでは，ファセット名はそれぞれのファセット値の上位概念を表す．例えばファセット名が「行事」というような抽象的な語に対して，そのファセット値は「祇園祭」や「花灯路」,「送り火」などファセット名をより具体的にしたものとなる．本研究では語同士の概念階層を表すファセットを対象として，ファセットの生成を動的に行い，そのファセット値をユーザが選択することで比較ページの検索を行う手法を提案する．

3.2 比較ページの検索に効果的なファセット

語同士の概念階層を表すファセットを生成するにあたり，比較ページの検索に効果的なファセットの要素として以下の 3 つが考えられる．

- ファセットの確からしさ
- ユーザのファセットに対する関心度
- ファセット値の数の多さ

ファセットの確からしさとはファセットを構成する語の概念階層の正しさ，つまりファセット名とファセット値が正しく上位下位関係をなす度合いとする．ファセットに対する関心度とはユーザがそのファセットを過去の Web ページの比較で用いてい

る度合いであり，例えば「祇園祭」や「葵祭」などのページをよく閲覧しているユーザは「祭り」という観点から Web ページを比較していると考えられ,「祭り」をファセット名に持つファセットの関心度は高いと考えられる．今回はユーザの Web 閲覧履歴を用いて，ファセットに対する関心度の推定を行う．またファセット値の数が多ければ比較ページの種類が多くなり，ナビゲーションの幅が広がると考えられる．本研究ではこの 3 つの要素が高いファセットを比較ページの検索に効果的なファセットと定義する．

4. 閲覧履歴を用いた動的なファセットの生成

本節では比較ページ検索に効果的なファセットを動的に生成する手法について述べる．本手法は主にファセットの生成とファセットのランキングという 2 つのステップで構成されている．ファセットの生成では閲覧中のページ cp と履歴ページ集合 RP を入力とし，ファセットの集合 $\mathcal{F} = \{F_1, F_2, \ldots, F_n\}$ が出力される．ここで，ファセット F_i はファセット名 h_i とファセット値の集合 $C_i = \{c_1^i, c_2^i, \ldots\}$ によって構成される．ファセットのランキングでは，入力としてファセット集合 $\mathcal{F}$ が与えられ，順序を考慮したファセット列 $rank(\mathcal{F}) = \{F'_1, F'_2, \ldots, F'_n\}$ が得られる．ただし，$F'_i = F_j (1 \leqq i, j \leqq n)$ である．

4.1 ファセットの生成

1 節で述べたように，本研究においてファセットを提示する意義は閲覧中のページと比較される Web ページを様々な観点から分類して検索できるようにすることにある．本研究では語同士の概念階層を表すファセット，つまりファセット名とファセット値が上位下位関係をなすファセットを生成するが，分類を行う上で互いのファセット値は同位関係であることが望ましい．もし閲覧中のページがあるファセットによって分類されると仮定すると，そのファセット名は閲覧中のページの特徴を表す語の上位概念にあたり，ファセット値は閲覧ページの特徴語の同位概念にあたると考えられる．そこで我々は，閲覧ページの特徴語に対する上位語と同位語を発見することでファセットを生成する手法を提案する．出力であるファセット集合 $\mathcal{F}$ を得るための手順は以下のとおりである．

（1） 閲覧ページ cp から特徴を表す語集合 T を抽出する．
（2） $t \in T$ の同位語と上位語を発見することで t に対応するファセット集合 $\mathcal{F}(t)$ を生成する．
（3） $F \in \mathcal{F}(t)$ のファセット名 h の下位語発見を行い，それを F のファセット値集合 C と比較することで，$\mathcal{F}(t)$ の検証を行う．不適だと判断されたファセットは $\mathcal{F}(t)$ から除かれる．
（4） 全ての $t \in T$ について (2) と (3) を繰り返し行い，全ファセット集合 $\mathcal{F} = \bigcap_{t \in T} \mathcal{F}(t)$ を得る．

4.1.1 閲覧ページにおける特徴語抽出

まず閲覧ページ cp の特徴ベクトル $\boldsymbol{s}_{cp}$ を生成する手法について検討する．文書からの特徴語抽出は以前から盛んに研究されているが，大きく分けて複数の文書を用いて特徴語の抽出を行う手法と単独の文書から特徴語を抽出する手法が存在する．TF-IDF をはじめとした複数の文書を用いて語の重み付けを行う手法やユーザの閲覧履歴を用いてキーワード抽出を行う手法 [15] などが提案されているが，本研究で抽出する特徴語とは Web ページのテーマを表すような語であり，他の Web ページによって影響されるとは考えにくい．今回は単独の Web ページから特徴ベクトルを生成する手法として，以下の 3 つの手法について検討する．

(TF) 単語の出現回数（TF）をもとにした特徴ベクトルの生成
(EL) HTML の要素を考慮した特徴ベクトルの生成
(TR) TextRank [16] を用いた特徴ベクトルの生成

なお本研究で特徴ベクトルの要素として用いるのは，MeCab(注1)を用いて Web ページ中のテキストを解析し名詞またはその連結と推定されたもののうち，あらかじめ用意したストップワードに含まれない語である．

(TF) は TF のみを考慮した特徴ベクトル生成手法であり，語 w に対し特徴ベクトル $\boldsymbol{s}_{cp}$ の w 成分にあたる $\boldsymbol{s}_{cp}(w)$ を以下のように定義する．

$$\boldsymbol{s}_{cp}(w) = tf(w)$$

ここで $tf(w)$ は文書中における w の出現回数である．

(EL) は HTML の要素によって重みを変えることで HTML の構造を考慮した特徴ベクトルを生成する手法である．これはタイトルや見出しに現れる語の方が本文中にしか現れない語よりも重要であることが多いという仮定に基づいている．特徴ベクトル $\boldsymbol{s}_{cp}$ の w 成分である $\boldsymbol{s}_{cp}(w)$ を以下のように定義する．

$$\boldsymbol{s}_{cp}(w) = \sum_{e \in EL} \alpha_e \cdot tf_e(w)$$

ここで，HTML 内に含まれる要素集合を EL とし，その中に含まれる任意の要素 e に対する重みを α_e，w の出現回数を tf_e とする．例えば <title> 要素の重みを 5，<body> 要素の重みを 1 のように設定することで，HTML の構造を考慮した特徴ベクトルが得られると考えられる．ただし，任意の要素に対し重みを決定しなければならず，パラメータの数が多くなってしまう点に注意が必要である．

(TR) はページ中の語に対し，TextRank を適用した結果の値からなる特徴ベクトルを生成する手法である．本手法ではページ中に含まれる語をノードとし，長さ N 語中での語同士の共起回数を枝の重みとした無向グラフに対して TextRank アルゴリズムを適用した．

特徴ベクトル $\boldsymbol{s}_{cp}$ の要素で，値の高い上位 n 語からなる語集合 T を特徴語の集合として抽出する．

4.1.2 同位語・上位語の発見

閲覧ページの特徴語 $t \in T$ に対して，同位語と上位語を発見することでファセット集合 $\mathcal{F}(t)$ を生成する手法について述べる．これまでの我々の研究 [13] [14] [17] では，ファセット生成においてまず同位語集合を発見し，その同位語集合を用いて共通の上位語を求めるというプロセスを行っていた．この手法は同位語発見と上位語発見のための言語パターンをそれぞれ用意

(注1)：http://mecab.sourceforge.net/

しているが，同位語の発見と上位語の発見を独立に行うことで同位語集合に異なる文脈のものが混ざりやすくなり，ファセット生成の精度が低下するという問題がある．例えば「阪神」という語の同位語を考えた場合，プロ野球の球団という上位語を文脈に考えれば「巨人」や「日本ハム」などの語が考えられるが，鉄道会社という上位語を文脈とすると「阪急」や「JR」といった語が挙げられる．もし同位語集合にこれらの語が混在すると，共通の上位語を求めることは非常に困難である．また発見した同位語集合から共通の上位語を求めるプロセスは多大な時間を費やし，実時間で処理することが求められる比較ページの検索には不向きである．

そこで本研究では，1 つの言語パターンから同位語と上位語を同時に発見し，異なる文脈に考慮したファセット生成手法を提案する．我々は複数の同位語とそれらの上位語によって表現されるフレーズの存在に着目した.「のような」や「などの」のような言語パターンは,「りんごやみかんのような果物」や「コーラやソーダなどの炭酸飲料」のようにパターンの前に具体的な語が列挙され，パターンの後にそれらを汎化した語が現れることが多い．もしパターンの前に列挙されている具体的な語に閲覧ページの特徴語 t が含まれていれば，それらの語は t の同位語，パターンの後に出現する語は t の上位語であると考えられる．これらの語をそれぞれファセット値集合 C と上位語 h として抽出し，ファセットを生成する．対象となるフレーズから同位語と上位語を切り出す手法は本節では詳しく述べないが，形態素解析は行わず全ての区切り方を記憶しておき，コーパスに含まれる全対象フレーズからの統計的な情報を基に最も適切な区切り方を推定している．本手法では t と言語パターンを含むクエリによる Web 検索結果のタイトルとスニペットを上位語発見のコーパスとして用いる．例えば，犬の同位語と上位語を発見したい場合は “犬などの” というクエリでフレーズ検索することで，犬の同位語と上位語を含むフレーズの多いコーパスの取得が期待される．このコーパスを走査することで同位語と上位語の組み合わせが複数得られ，ファセット集合 $\mathcal{F}(t)$ が生成される．

4.1.3 ファセットの検証

言語パターンを用いて閲覧ページの特徴語 t の同位語と上位語を発見することでファセット集合 $\mathcal{F}(t)$ が得られたが，その中の全てのファセットが必ずしも上位下位関係を成しているとは限らない．例えばコーパス中に「ペンギンやパンダなどの動物のこどもは」と「ペンギンやパンダなどの世界中の動物が大集合」というフレーズが含まれていたとする．ペンギンの上位語を発見する場合，前者からは「などの」という言語パターンの直後にある「動物」を抽出すればよいが，後者からパターンの直後にある「世界中」が誤って抽出されてしまう．また形態素解析を用いて修飾句を無視するようにしたとしても，後者からは正しく抽出できるが前者からは「こども」という語が誤って抽出されてしまう．このように言語パターンを用いて上位語を確実に発見することは困難である．ここでもしファセット F が同位語・上位語発見によって正しく生成されたのであれば，そのファセット名 h とファセット値 C_i は上位下位関係にあたるはずである．そこでファセット名の下位語発見を行い，ファセット値集合 C と比較することで生成したファセットの検証を行う．

上位語発見のための言語パターンは，与えられる元の語と抽出対象となる語の役割を入れ替えることで下位語発見にも利用することができる．例えば炭酸飲料の下位語を発見したい場合,「コーラやソーダなどの炭酸飲料」といったフレーズの「などの」の前を見ることでコーラやソーダなどの下位語が得られる．ここであるフレーズ l に対するファセット名 h の下位語集合を $Hyponym(h,l)$ と定義する．もし $Hyponym(h,l)$ に含まれる語とファセット値集合 C に含まれる語に共通部分があれば，h と C は上位下位関係にあると考え，そのファセットは妥当であると判断する．またこのとき，C には含まれていないが $Hyponym(h,l)$ には含まれている語を新たにファセット値として C に加える．これは $Hyponym(h,l)$ が共通の上位語 h を持ち，かつ C と共通部分を持つため同じ文脈における同位語であると考えられるからである．例えば，2 つの同位語集合 { 哺乳類，魚類，両生類，鳥類，爬虫類 } と { ペンギン，クジラ，カエル，ワシ，トカゲ } がある場合，どちらの集合の語も「動物」という共通の上位語を持つが，概念階層における層の深さが異なるためこの 2 つの集合を 1 つの同位語集合であるとみなすことはできない．下位語発見のコーパスは上位語発見のコーパスと同様，ファセット名 h と言語パターンを含むクエリによる Web 検索結果のタイトルとスニペットを用いる．例えば，動物の下位語を発見したい場合は “などの動物” というクエリでフレーズ検索することで，動物の下位語表現を含むフレーズの多いコーパスの取得が期待される．このコーパスに含まれる全対象フレーズに対してこの作業を行うことでファセットの妥当性を検証し，妥当であると判断されたもの以外は $\mathcal{F}(t)$ から取り除かれる．この同位語・上位語発見とその検証という一連の手順を全ての特徴語について行い，得られたそれぞれファセット集合の和集合が最終的な出力である $\mathcal{F}$ である．

4.1.4 文脈語の推定とファセット生成への利用

これまでの手順で閲覧ページのある特徴語 t からファセットが生成される手法を説明した．しかし，このファセット生成手法は t のみからファセットが生成され，ユーザがどのような意図でこの Web ページを閲覧しているかなどは一切考慮されていない．例えば，ある Web ページから「八坂神社」という特徴語が得られた場合，そこから得られるファセットの 1 つとしてファセット名が「観光名所」でファセット値が「金閣寺」や「銀閣寺」,「知恩院」などで構成されるファセットが考えられる．しかし，もしユーザが「花灯路」に関する情報を求めて現在のページを閲覧しているのなら，ファセット値として「花灯路」が行われる観光名所である「高台寺」や「清水寺」を提示するべきであろう．このようなユーザが一貫して関心を持っている話題は，Web ページの閲覧履歴を調べることで推定できると考えられる．本研究ではユーザが一貫して関心を持っている話題を表す語を文脈語と呼び，文脈語をユーザの閲覧履歴から推定することでよりユーザ個々に合わせたファセットの生成を行う．文脈語はその特徴から閲覧中のページを含む全履歴ペー

ジを通して重要な語であり，さらに履歴ページ中の多くのページに含まれるような語であると考えられる．ある語 w における Web 閲覧の文脈としての重要度を以下の式で定義する．

$$\mathrm{CTX}(w) = \frac{\sum_{p \in P} \boldsymbol{s}_p(w)}{|P|} \cdot df(w, P)$$

ここで P は閲覧中のページを含む全履歴ページの集合，$\boldsymbol{s}_p(w)$ はページ p の特徴ベクトルの w 成分，*df(w,P)* はページ集合 P における w を含むページの割合を示す．CTX(w) の高い上位 m語を文脈語として抽出する．

抽出した文脈語を利用して，よりユーザの Web 閲覧の文脈にあったファセットを生成するために，上位語発見のコーパスと下位語発見のコーパスの取得方法を改良する．コーパス取得のためのクエリに文脈語を加えることで，文脈語とは関連のない Web ページのタイトルとスニペットがコーパスに現れにくくなる．例えば文脈語として「花灯路」が得られた場合，特徴語である「八坂神社」の上位語発見のコーパスを取得するためのクエリの例は「“八坂神社などの”　花灯路」であり,「花灯路」とは関連のない「金閣寺」や「銀閣寺」がコーパスに現れる可能性は低くなると考えられる．

4.2 ファセットのランキング

ここでは生成されたファセット $F_i \in \mathcal{F}$ の重要度を 2 節で述べた比較ページの検索に効果的なファセットに必要な要素に基づいて算出し，重要度の高い順に並べ替えられたファセット列 $rank(\mathcal{F})$ を得る手法について述べる．

4.2.1 ファセットの確からしさ

本研究におけるファセットの確からしさとは，ファセット名とファセット値が正しく上位下位関係をなす度合いを指す．ファセットの確からしさを以下の式で定義する．

$$\mathrm{Probability}(h_i, C_i) = \mathit{fnf}(h_i) \cdot \sum_j \mathit{cooccur}(h_i, c_j^i)$$

fnf(h_i) は F_i のファセット名 h_i が上位語発見のコーパスと下位語発見のコーパスを通じて対象フレーズに出現した回数を表す．また *cooccur*(h_i, c_j^i) はファセット名 h_i とファセット値 $c_j^i \in C_i$ が上位語発見のコーパスと下位語発見のコーパスを通じて対象フレーズで共起した回数を表す．例えばファセット名が「行事」，ファセット値が「祇園祭」「花灯路」「送り火」からなるファセットを想定し，対象フレーズとして「祇園祭や花灯路などの行事」と「祇園祭や送り火などの行事」が得られたとすると，*fnf*(h_i) は 2，*cooccur*(h_i, c_j^i) は 4 となる．*fnf*(h_i) が高いことは h_i が c_j^i の上位語としてコーパス中に頻繁に出現していることを意味し，$\sum_i \mathit{cooccur}(F_i)$ が高いことは多くのファセット値が h_i の下位語としてコーパス中に出現していることを意味している．よって，Probability(h_i, C_i) がファセット名とファセット値が正しく上位下位関係をなす度合いを表すといえる．

4.2.2 ファセットに対する関心度

ファセットに対する関心度とはユーザがそのファセットを過去の Web ページの比較で用いている度合いを表す．ファセットに対する関心度を以下の式で定義する．

$$\mathrm{Interest}(C_i, RP, cp) = \frac{d(C_i, RP, cp)}{|RP|}$$

ここで $|RP|$ は閲覧履歴に含まれるページ数，$d(C_i, RP, cp)$ は RP の中で，閲覧ページ cp の特徴ベクトルに含まれていないファセット値集合の少なくとも一つが含まれるページの数である．端的に述べるとファセット値のいずれかが履歴ページに含まれている割合が多いほどそのファセットに対する関心度は高くなる．

4.2.3 ファセットの重要度の算出

ファセットの重要度を以下の式で算出する．

$$\mathrm{Score}(F_i) = n(C_i) \cdot \mathrm{Probability}(h_i, C_i) \cdot \mathrm{Interest}(C_i, RP, cp)$$

ただし $n(C_i)$ はファセット値の数を表す．ファセット集合 $\mathcal{F}$ に含まれるすべてのファセットに対して上の式で重要度を算出し，重要度の高い順に並べ替えることでファセット列 $rank(\mathcal{F})$ が得られる．

5. 実　　験

本手法では閲覧中の Web ページから特徴語を抽出し，それぞれの特徴語ごとに上位語・同位語・下位語発見手法を利用することでファセットの生成を行った．特徴語抽出手法の評価とファセット生成手法の評価を行うために 2 種類の実験を行った．

5.1 特徴語の抽出精度を測る実験

3 節で述べた 3 つの特徴ベクトル生成手法に対して，特徴語抽出の精度を測る実験を行った．Web サイトのトップページのような比較的テキストの少ない Web ページ 5 件と Wikipedia(注2) のようなテキストの多い Web ページ 5 件に対し，各手法で特徴ベクトルの要素の値が高い上位 3 件，5 件，10 件の適合率を人手で評価した．なお，今回はテキスト中の総名詞数が 1000 語未満の Web ページをテキストの少ない Web ページ，総名詞数が 1000 語以上の Web ページをテキストの多い Web ページとして評価を行った．また (EL) における HTML の各要素 e に対する重み α_e については，<title> 要素に対する重みを 5，<meta> 要素の keyword 属性と description 属性に対する重みを 5，<img> 要素の alt 属性に対する重みを 3，<h1> 要素に対する重みを 4，<h2> 要素に対する重みを 3，<h3> 要素に対する重みを 2 とした．表 1 と表 2 に，それぞれ使用した Web ページの URL の一覧と各手法における @k(上位 k 件の適合率) を示す．

この結果を見てわかるとおり，Web サイトのトップページのようなテキストの少ない Web ページでは (EL) が，テキストの多い Web ページでは (TR) が最も精度よく特徴語を抽出できていた．

これは Web サイトのトップページが物語のような文脈のあるテキストではなく，訪問者が注目するような情報を構造化した形式，例えば箇条書きのような形で書かれていることが多いため，そもそもテキストが少ない上にそのような構造を考慮せずに語同士の共起を利用する (TR) はうまく特徴語を抽出でき

(注2)：http://ja.wikipedia.org/wiki/

表 1　使用した Web ページの URL の一覧

	URL	総名詞数
テキストの少ない Web ページ	http://cweb.canon.jp/camera/dcam/index.html	541
	http://www.nttdocomo.co.jp/	255
	http://www.tokyodisneyresort.co.jp/top.html	514
	http://www.softbankhawks.co.jp/	440
	http://www.harborland.co.jp/	327
テキストの多い Web ページ	http://ja.wikipedia.org/wiki/鹿苑寺	1983
	http://ja.wikipedia.org/wiki/USJ	5665
	http://ja.wikipedia.org/wiki/スマートフォン	5052
	http://ja.wikipedia.org/wiki/ミッキーマウス	2593
	http://ja.wikipedia.org/wiki/オーロラ	2250

表 2　上位 k 件における各手法の平均適合率

	手法	@3	@5	@10
テキストの少ない Web ページ	(TF)	66.7%	60.0%	44.0%
	(EL)	**86.7%**	**88.0%**	**70.0%**
	(TR)	60.0%	52.0%	44.0%
テキストの多い Web ページ	(TF)	80.0%	68.0%	60.0%
	(EL)	80.0%	68.0%	54.0%
	(TR)	**86.7%**	**76.0%**	**72.0%**

表 3　テキストの少ない Web ページからの特徴語の抽出例

(TF)	(EL)	(TR)
B 組	ヤフードーム	ホークス
ヤフードーム	福岡ソフトバンクホークス	ホークス情報
ホークス	オフィシャルサイト	選手
ケータイ	ホークス	チケット
春季キャンプ	ケータイ	チケット情報

表 4　Wikipedia「スマートフォン」エントリからの特徴語の抽出例

(TF)	(EL)	(TR)
スマートフォン	スマートフォン	製
OS	OS	OS
Windows	Windows	スマート
Android	搭載	搭載 OS
搭載	Android	搭載

なかったと考えられる．表 3 はソフトバンクホークスのオフィシャルサイトトップページから特徴語候補の上位 5 件を抽出した結果である．(EL) が球団名や本拠地の名前を抽出できているのに対し，(TR) はうまく抽出できていないことがわかる．

逆にテキストが多い Web ページでは (TR) が比較的有効であった．適合率で他の手法と最も顕著な差が表れた Wikipedia の「鹿苑寺」のエントリは，金閣寺の歴史や構造などの説明がきちんとした文章でなされており，全体の文脈を考慮した特徴語抽出ができたと思われる． 表 4 は Wikipedia の「スマートフォン」エントリから抽出された上位 5 件の特徴語である．Wikipedia の「スマートフォン」のエントリは，総名詞数が 5052 語とテキストが非常に多いにも関わらず (TR) は他の手法よりもうまく抽出できていなかった．これは「鹿苑寺」のエントリとは異なり，このエントリの半分ほどがスマートフォンを販売しているメーカーやその OS，機種などの箇条書きによる説明で占められており，エントリ自体の文脈が薄いため，(TR) の精度が悪くなったと考えられる．このように (EL) と (TR) にはそれぞれ特徴があるので，テキストの長さや文章の構造を考慮することでどのような Web ページにもロバストに特徴語を抽出できるよう改善する必要がある．

表 5　単独の語から生成したファセットの評価結果

	ファセットの適合率	ファセット値の平均適合率	適合ファセット値の最大数
ミッキーマウス	100.0%	100.0%	5
Ruby	50.0%	100.0%	5
サッカー	50.0%	100.0%	9
チワワ	100.0%	100.0%	13
京都大学	33.3%	81.8%	9
平均	66.7%	96.4%	8.2

5.2　ファセット生成の精度を測る実験

今回は単独の語から生成されるファセットの精度と，テストセットとして用意された Web ページの閲覧履歴と閲覧中の Web ページから提示されたファセットの精度を測る実験を行った．ファセットの精度を評価する指標として，本研究ではファセットの適合性とファセット値の適合性を以下のように定義する．

（1） ある Web ページの特徴語から生成されたファセット F_i のファセット名 h_i とファセット値集合 C_i の少なくとも 1 つが上位下位関係となっていれば，F_i はその Web ページに対するファセットとして適合している

（2） 適合しているファセット F_i に対して，ファセット値 c_j^i がファセット名 h_i の下位語であれば c_j^i は F_i に適合している

上位語・同位語・下位語発見のためのコーパス取得には Yahoo! Search BOSS API(注3) を使用し，得られた検索結果の上位 50 件のタイトルとスニペットをコーパスとした．またコーパス取得のクエリに接続する言語パターンには予備実験で最も精度のよかった「などの」を用いた．

まず単独の語から生成されるファセットについての評価を行った．今回使用した語は「ミッキーマウス」「Ruby」「サッカー」「チワワ」「京都大学」の 5 つである．表 5 は各語に対するファセットの適合率とファセット値の平均適合率，適合したファセット値の最大数を示している．ファセットの適合率，ファセット値の平均適合率ともに良い結果となっており，精度よくファセットが生成されているといえる．特に適合したファセット値の数が多いことがわかるが，これは上位語・同位語発見のコーパスと下位語発見のコーパスの両方からファセット値を抽出するこの提案手法がうまく働いていることを示している．

次に実際の Web 閲覧を想定し，閲覧中の Web ページに提示されるファセットの精度を測る実験を行った．ユーザがある話題に一貫して興味を持って Web ページの閲覧を行った場合を想定し，Web ページの閲覧履歴と閲覧中の Web ページをテストセットとして用いる．このテストセットを用いて，提案手法が

(注3)：http://developer.yahoo.com/search/boss/

表 6　テストセット

	テストセット 1	テストセット 2	テストセット 3
関心のある話題	ディズニーランド	花灯路	野球
閲覧ページ	ディズニーランド ミッキーマウス	八坂神社 花灯路	ソフトバンク 野球
履歴ページ 1	ディズニーランド ミニーマウス	高台寺 花灯路	日本ハム 野球
履歴ページ 2	ディズニーランド スティッチ	清水寺 花灯路	巨人 野球

表 7　各テストセットにおける適合度

	閲覧履歴を利用	閲覧履歴を利用しない
テストセット 1	1	1
テストセット 2	2	0
テストセット 3	2	1
合計	5	2

表 8　各テストセットにおけるファセット値の平均適合率

	閲覧履歴を利用	閲覧履歴を利用しない
テストセット 1	100.0%	51.3%
テストセット 2	100.0%	0
テストセット 3	100.0%	75.0%
平均	100.0%	42.1%

ユーザの関心をファセットのランキングに反映できるのかを調査した．今回は簡単のため Web ページをその特徴語で表現し，特徴語集合からなるテストセットを 3 セット用意した．閲覧履歴は閲覧直前の 2 つの Web ページを対象とし，その特徴語集合で表現されている．表 6 にそれぞれの関心を持っている話題と実際にテストセットに含まれている特徴語を示す．この実験ではファセットの適合性をユーザが関心のあるファセットとそうではないファセットで区別するため，ファセットの適合度に応じて点数を付与することとする．もし最上位がユーザの関心のあるファセットであれば 2 点，そうではない適合ファセットであれば 1 点，適合していないファセットであれば 0 点を与える．表 7 と表 8 にそれぞれ閲覧履歴を用いた場合と用いない場合の，各テストセットにおける適合度とファセット値の平均適合率を示す． このように，閲覧履歴を利用した方が適合度およびファセット値の平均適合率ともによい結果が得られた．テストセット 2 では，ユーザは「花灯路」に関心を持ちながら,「花灯路」が行われる場所の情報を求めて Web ページを閲覧しているというケースを想定している．閲覧履歴を用いた場合では「寺院」という名のファセットが得られ，ファセット値も「花灯路」が行われる八坂神社や知恩院，高台寺，清水寺が得られており，本手法が有効に働くことが確認された．閲覧履歴を用いない場合では「観光」というファセット名が得られたが，八坂神社の上位語としては適切ではない．これは本来ファセット名を取得するためのコーパスでは「観光名所」として表現されているフレーズも存在するのだが,「観光地」や「観光スポット」と表現されているフレーズも数多くあるため，ファセット名抽出の際に誤って「観光」で区切ってしまうのである．本手法ではフレーズからファセット名とファセット値を抽出する際に出現単語の頻度に基づいて区切り方を推定しているため，このように誤った区切り方をしてしまい，ファセットの適合性の低下の要因となっているので改善の余地がある．

また「などの」という言語パターンにも少なからず問題が存在する.「などの」の前後に出現する語には上位下位関係が成り立つことも多いが,「日本やアメリカなどの大学」というフレーズのように「大学」を修飾するための名詞を並列するために「などの」という助詞が用いられることがある．これは「の」という助詞に複数の役割があり，上位下位関係を表す場合の「の」は同格を表す役割を担っている．このように助詞である「の」の役割を判定することで，より精度の向上が見込めると考えられる．

6. Web 検索エンジンを用いた比較ページの検索

4 節では Web 閲覧履歴を用いて比較ページ検索に効果的なファセットの生成を行う手法について述べた．本研究の最終的な目標は，提示されたファセットからユーザが興味のあるファセットとそのファセット値を選択した場合に，閲覧ページと比較されるページの検索を支援することである．本節ではユーザが選択したファセットとファセット値から，閲覧ページと比較されるようなページを Web 検索エンジンを用いて検索する手法について述べる．入力はユーザが選択したファセットと閲覧ページの特徴語，出力は比較ページの候補となる Web ページ集合である．入力で得られた語を用いて，比較ページ検索のためのクエリを生成し，適切な比較ページ候補となる Web ページ集合を取得することを目指す.

6.1 本研究で検索する比較ページ

本研究で検索される比較ページを以下のように定義する.

（1） 比較されるファセット以外の性質は閲覧ページにおける性質と同一のものである

（2） 選択されたファセット値についてより詳しく述べられている.

本手法ではユーザに明示的にファセットとそのファセット値を選択させることで，比較ページへのナビゲーションを行う．もしナビゲーションの過程で，比較されるファセット以外の性質を変えてしまうと比較する軸が曖昧なものとなってしまい，ファセットに応じた比較ページへのナビゲーションが保証されなくなってしまう危険性がある．よって今回は，比較されるファセット以外の性質は閲覧ページにおける性質と同一のものであるような比較ページの検索を行う.

(2) に関しては，ユーザが興味のあるファセットを選択する際に他の数多くのファセット値を選ばず，そのファセット値を選択したということは，他のファセット値に関する情報よりも選択したファセット値に関する情報をよく知りたいという意思表示であると考えることができる．そこで選択されなかったファセット値に関する情報より，選択したファセット値に関する情報について詳しく述べられた Web ページ集合の検索を行う.

6.2 比較ページ検索のためのクエリ生成

本研究ではキーワード検索型の Web 検索エンジンを用いて，閲覧ページと比較されるようなページの検索を行う．ここでは前節で定義された比較ページ集合を取得するためのクエリの生

成手法について述べる．まず検索される比較ページ集合には，選択されたファセット F 以外の閲覧ページの特徴を持っていなければならないので，閲覧ページの特徴語から選択されたファセットに含まれるファセット値を取り除いた語集合 T_{ex} がクエリに必要である．

また，選択したファセット値 c_{sel} に関する情報について詳しく述べられた Web ページ集合を検索するにはそのファセット値を表す語をクエリとして検索すればよいが，その検索結果には選択されなかったファセット値に関する Web ページや，まとめページのような，選択したファセット値と選択されなかったファセット値に関する情報が広く浅く書かれたような Web ページも含まれる．そこで選択されたファセット値に関する情報のみが出現するように，選択されなかったファセット値に関しては NOT 検索を行う．

まとめると，T_{ex} と c_{sel} と選択されなかったファセット値集合 $C - c_{sel}$ の否定を AND で連結したものがクエリとなる．

例えば，閲覧ページの特徴語が「八坂神社」「花灯路」「ライトアップ」で，ファセット値に「八坂神社」と「清水寺」と「高台寺」を持つファセットの「高台寺」をユーザが選んだら，生成される比較ページ検索のためのクエリは「花灯路 AND ライトアップ AND 高台寺 AND -八坂神社 AND -清水寺」となる．最終的な出力はこのクエリで Web 検索を行った検索結果集合である．

7. ま と め

本研究では閲覧中の Web ページに複数のファセットを動的に生成する手法を提案した．閲覧中の Web ページを表す複数の特徴語に対して，上位語と同位語を発見することでそれぞれファセットを動的に生成し，さらに下位語発見を行うことで生成したファセットの検証を行った．ファセット生成の精度を測る実験では，Web 閲覧履歴を利用することでよりユーザの関心に沿ったファセットを提示できることがわかった．さらに生成されたファセットから興味のあるものを選択することで，選択されたファセットに沿った比較ページの検索についても提案した．比較ページの検索については今後実験を行い，提案した手法が有効であるかどうか確かめていきたいと考えている．

謝辞 本研究の一部は，京都大学 GCOE プログラム「知識循環社会のための情報学教育研究拠点」，および，文部科学省科学研究費補助金（課題番号：18049041，21700105），および，独立行政法人情報通信研究機構の高度通信・放送研究開発委託研究「電気通信サービスにおける情報信憑性検証技術に関する研究開発 課題ア Web コンテンツ分析技術」（研究代表者：田中克己）によるものです．ここに記して謝意を表します．

文 献

[1] T. Yoshida, S. Nakamura and K. Tanaka: “WeBrowSearch: toward web browser with autonomous search”, Web Information Systems Engineering–WISE 2007, pp. 135–146 (2007).

[2] D. Tunkelang: “Dynamic category sets: An approach for faceted search”, ACM SIGIR Workshop on Faceted Search (2006).

[3] C. Li, N. Yan, S. Roy, L. Lisham and G. Das: “Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia”, Proceedings of the 19th international conference on World wide web, pp. 651–660 (2010).

[4] S. Lim, Y. Liu and W. Lee: “Faceted search and retrieval based on semantically annotated product family ontology”, Proceedings of the WSDM’09 Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 15–24 (2009).

[5] D. Dash, J. Rao, N. Megiddo, A. Ailamaki and G. Lohman: “Dynamic faceted search for discovery-driven analysis”, Proceeding of the 17th ACM conference on Information and knowledge management, pp. 3–12 (2008).

[6] S. Basu Roy, H. Wang, G. Das, U. Nambiar and M. Mohania: “Minimum-effort driven dynamic faceted search in structured databases”, Proceeding of the 17th ACM conference on Information and knowledge management, pp. 13–22 (2008).

[7] K. Yee, K. Swearingen, K. Li and M. Hearst: “Faceted metadata for image search and browsing”, Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 401–408 (2003).

[8] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen and K. Yee: “Finding the flow in web site search”, Communications of the ACM, pp. 42–49 (2002).

[9] G. Miller: “WordNet: a lexical database for English”, Communications of the ACM, **38**, 11, pp. 39–41 (1995).

[10] M. Hearst: “Automatic acquisition of hyponyms from large text corpora”, Proceedings of the 14th International Conference on Computational linguistics, pp. 539–545 (1992).

[11] Z. Ghahramani and K. Heller: “Bayesian sets”, Proceedings of the 19th Annual Conference on Neural Information Processing Systems, pp. 435–442 (2005).

[12] K. Shinzato and K. Torisawa: “A simple www-based method for semantic word class acquisition”, AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4, **292**, p. 207 (2007).

[13] H. Ohshima, S. Oyama and K. Tanaka: “Searching coordinate terms with their context from the web”, Web Information Systems–WISE 2006, pp. 40–47 (2006).

[14] H. Ohshima and K. Tanaka: “Real time extraction of related terms by bi-directional lexico-syntactic patterns from the web”, Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, pp. 441–449 (2009).

[15] 松尾豊, 福田隼人, 石塚満：“ユーザ個人の閲覧履歴からのキーワード抽出によるブラウジング支援”, 人工知能学会論文誌, **18**, 4, pp. 203–211 (2003).

[16] R. Mihalcea and P. Tarau: “TextRank: Bringing order into texts”, Proceedings of EMNLP, pp. 404–411 (2004).

[17] 川野悠, 大島裕明, 田中克己：“クエリに応じたファセットの動的抽出による Web 画像検索結果の提示”, 第 2 回データ工学と情報マネジメントに関するフォーラム (DEIM2010) 講演論文集 (2010).