Webページ集合を解とする全容検索

湯本高行[†] 田中克己[††]

知りたい情報について知識がない状態で検索を行う場合，ユーザは検索結果を閲覧しても，必要なすべての情報を得られたのかどうかを判断することができない．また，現在のページごとの検索では知りたい事柄について1ページで十分な情報を持ったページが存在するとは限らず，そのため解として適切なページが見つかるとは限らない．そこで，ユーザの求める情報の全容を表すページ集合を発見する全容検索を提案する．全容検索は，通常のページごとの検索結果から，あるキーワードについて話題の広さと深さの両方を兼ねそなえたページ集合を生成し，それをランキングする．全容検索では，話題の漏れのないようにページを網羅的に収集するのではなく，検索結果集合から詳細グラフという語の詳細関係を表すグラフを計算し，ページ集合の表す内容やページ間の内容の重複を表現し，ページ間の内容の重複ができるだけ少なく，ユーザが効率良く閲覧できるようなページ集合を生成する．本稿では，ページ集合を対象とした全容検索と通常の検索やページごとの全容検索などを比較し，ページ集合を解として全容検索をすることの有効性を示す．

Overview Search Returning Web Page Sets as Answers

Takayuki Yumoto[†] and Katsumi Tanaka[††]

When a user searches Web by a query keyword X about which he/she has no knowledge, it is difficult for him/her to evaluate to what extent each answer page includes topics about X. Furthermore, conventional page-by-page search might not always return an appropriate page as an answer that include enough topics about X. In order to solve these problems, we propose overview search to find page sets which describe overview about what users want to know. Overview search is to find page sets which describe both of wide topics and deep detail about a given query and rank them. However, users don't want to browse too big page sets. Therefore, in overview search, pages in a page set should have less duplicated information. We construct as compact page set as possible by using a Detailing Graph. The Detailing Graph represents detailing-relationship between terms in search results. We express the information which page sets have and the duplication between pages by using the Detailing Graph and construct the page sets which have more information and less duplication. In this paper, we compare our overview search returning page sets as answers, overview search returning pages as answers and conventional Web search. We show some experimental results and the effectiveness of overview search.

1. はじめに

近年，インターネットの普及や検索技術の向上により，ユーザの求める情報が以前より容易に見つけられるようになっているが，それがあてはまらない場合がある．Taylorは情報要求を「直観的要求」,「意識された要求」,「形式化された要求」,「調整済みの要求」の4階層に分類した[1]．クラスタリングの利用[2]などの検索における新たな試みは要求を形式化することができて初めて役に立つものであり，要求を意識しているが，形式化できていない場合，たとえば，検索したい事柄について詳しい知識がない場合には検索はいまだ難しいものである．たとえば,「鳥インフルエンザ」について知りたい場合，感染源や予防策，日本での症例，海外での症例などさまざまな側面があり，まったく知識がない人にとってはまず何を探してよいか分からない．ユーザはとりあえず,「鳥インフルエンザ」というクエリで検索を始めるが，少数のページだけを閲覧して，ある程度分かったつもりになり，他の重要な情報

† 兵庫県立大学大学院工学研究科電気系工学専攻
Department of Electrical Engineering and Computer Sciences, Graduate School of Engineering, University of Hyogo
†† 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Graduate School of Informatics, Kyoto University

本研究は京都大学大学院情報学研究科社会情報学専攻に所属している際に行われた．

に気付くことなく，一部についてより詳しい内容を探し始めてしまう場合がある．この場合，重要な情報を得る機会を逸してしまっている．このような状況を避けるためには，現状ではユーザはつねに大量のページを閲覧するしかなく，これはユーザにとって大きな負担である．このような状況を改善するためには，検索結果のページ一覧からどのページを閲覧すれば，検索結果の全容をつかむことができるのかをユーザに示すことが有効である．

そこで本研究では，検索結果集合から複数のページを組み合わせ，全容を示すページセットを生成する手法「**全容検索**」を提案する．全容検索では，ただ漏れのないようにページを網羅的に収集するのではなく，語の詳細関係を表した詳細グラフをもとにページセットが表す内容やページ間の内容の重複を考慮し，ユーザがより効率的に閲覧ができるように，できるだけ内容的重複の少ないページセットを生成する．

本稿の構成は以下のとおりである．2 章では，本研究の位置付けを関連研究とともに説明する．3 章では，本研究の対象であるページセットの定義について述べ，4 章では，それに基づいたページセットのランキングアルゴリズムについて述べる．5 章では，従来のページごとの検索と本研究の手法を比較して優位性を示し，6 章では，まとめと今後の課題について述べる．

2. 全容検索

2.1 位置付け

従来，多くの検索では，検索単位をページもしくは文にすることが大半であった．それに対して，筆者らは従来の検索単位を複数組み合わせた統合型検索（Page Set Ranking）を提案している[3),4)]．統合型検索は，ページセットを検索の単位として扱うことが最大の特徴であり，「ページセットが表現する内容」と「ページ間の関係」によってページセットを評価する．また，統合型検索では，入力は検索結果ページのランクつきリスト（現在の手法ではランクなしの集合でもよい）であり，出力はページセットのランクつきのリストである．**図 1** に統合型検索のイメージを示す．

統合型検索の目的はさまざまであるが，本研究では全容検索☆を対象とする．与えられたページの集合から話題の広さと深さを両立したページセットを生成し，ランキングする検索手法である．全容検索は，たとえばまったく知識がない分野についてのサーベイを行う場合などに有効である．これに対して，クラスタリングを利用する場合，クラスタリング結果の構造がユーザにとってある程度の助けになる可能性もある．しかし，これはユーザがある程度の予備知識を持っている状態では判断材料になるかもしれないが，知識のない状態では有効に使うことはできない．また，各クラスタ内のページはつねにそのクラスタに分類されるページで述べられている内容の一部しか述べていない可能性がある．そのため，ユーザが全容を理解するうえでどのページを閲覧すべきかをページセットとして提示することによって，ユーザは効率良く全容を理解することができる．全容検索はすべての内容を網羅するわけではなく，ユーザの理解の第 1 歩として情報を提示するものである．したがって，全容検索によって提示されたページセットを閲覧して全容を理解した後に，クラスタリングによる分類を用いて，さらに他のページを閲覧したり，また，別のキーワードを付加して検索を行ったりするなどが考えられる．さらに，全容検索ではページ間の内容の重複はなるべく避け，効率の良い閲覧を支援することも重要である．

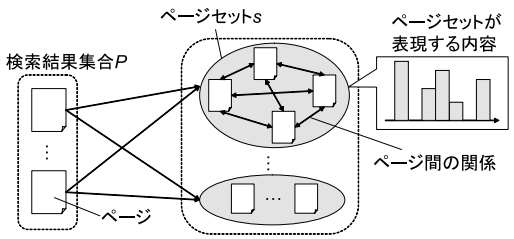

図 1 統合型検索のイメージ
Fig. 1 Image of Page Set Ranking.

かつての情報検索はクエリへの類似度などの内容分析のみによっていたが，現在の Web 検索エンジンはそれに加えて Google の PageRank[5)] のようなリンク分析などの手法も合わせることにより，検索精度を向上させることに成功している．我々はまず，内容分析のみによるアプローチをとり，ページペアに関するランキング[3)] を提案した．Sun らは与えられた 2 つのキーワードに関して比較可能なページをそれぞれ発見し，ページペアをつくる CWS[6)] を提案している．我々が提案したページペアランキングとの違いは，CWS は比較するためにページを集めるが，ページペアランキングでは比較したページを集め，視点の違うものどうしでペアを作成するというものである．また，我々はさらにランキングの対象を一般のページセットにも拡張した[4)]．

他の研究と本研究との違いはランキングに用いる単位がページではなく，ページセットであることである．また，リンク分析の手法の導入については今後考える

☆ 文献 4) で overview query と呼んでいたものである．

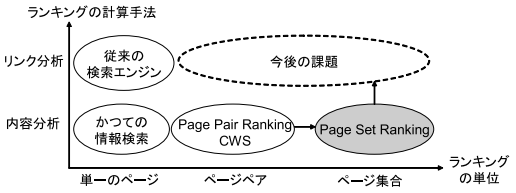

図 2 本研究と他の研究との関係
Fig. 2 Relationship between our research and other research.

べき課題である．**図 2** に本研究と他の研究との関係を示す．

2.2 関連研究

Cutting らは効率の良い文書の発見のためにクラスタリング手法を導入した[7]．このクラスタリングを Web の検索エンジンにも利用したサービスも存在し，それらのサービスでは検索結果をクラスタリングし，クラスタリングごとに検索結果を出力し，クラスタの階層構造とともに出力している[2]．このような手法では Web ページの分類はできるが，ページ間の関係が分からないうえ，たとえ，各クラスタから 1 つずつページを選んで閲覧することによって全容を理解できる可能性があっても，ユーザにはどれを閲覧すると最も効率的に全容を理解できるか分からない．

Tajima らはリンクでつながった Web ページやネットニュースやメールのスレッドに対して，検索の解として適切な範囲を抽出する手法を提案している[8]．また，風間らは検索結果をドメインやリンク構造を用いて，ページグループ，サイトグループとして集約して提示する手法を提案している[9]．これらの手法はリンクなどによってすでに関連づけられているものに対して解の粒度を大きくして適した解を発見しようというものであるが，提案手法は粒度を大きくするだけではなく，関連づけられていない任意のコンテンツを組み合わせて適した解を構成するものであり，より適した解を発見できる可能性がある．

Toda らは，検索結果ページの間の類似度によって仮想のリンクを生成し，それに対して PageRank のアルゴリズムを適用することで，概要的なページとそれを補完するページを発見する手法を提案している[10]．本研究の手法では，補完するページ間の内容的な関係も重視しているが，Toda らの研究ではその点については考慮されていない．

また，オントロジによって，探すべきトピックを決定するというアプローチもある[11] が，本研究では検索結果から生成するため，より多様なユーザの要求に応えることができる．

Oyama らは Web の検索件数から与えられた語の詳細語か否かを判定する手法を提案している[12]．Oyama らの手法では，以下の式が成り立っているとき語 B は語 A の詳細語と見なす．

$$\frac{DF(intitle(A) \wedge B)}{DF(intitle(A))} > \frac{DF(A \wedge B)}{DF(A)} \qquad (1)$$

ただし，$DF(intitle(A) \wedge B)$ は語 A をタイトル部分に含み，語 B も含むページの検索件数，$DF(intitle(A))$ はタイトル部分にキーワード A を含む検索件数，$DF(A \wedge B)$ は語 A と語 B を含むページの検索件数，$DF(A)$ は語 A を含む検索件数である．また，語 B が一般的な語である場合を排除するために式 (1) の統計的な正しさを χ^2 検定によって確認できたもののみ詳細語と見なしている．**表 1** に χ^2 検定に用いる 2×2 分割表を示す．表 1 の (a)，(b)，(c)，(d) が計算できれば，他の値も計算できるので，χ^2 検定を行うことができる．このため，同じ語に対して，n 個のキーワードが詳細語かどうか判定するためには，$2 + 2n$ 回の検索エンジンへの問合せが必要となる．本研究では，一般的な語を除去するためにこのアルゴリズムを利用している．

3. ページセット

3.1 定　　義

ページセットは単一のページもしくは複数のページからなる検索単位である．本稿では，一般的なページの集合と区別するために，検索単位となるようなページの集合をページセットと呼び，s などの小文字で表し，検索単位とならない集合は，検索結果集合など「○○集合」と表記し，P など大文字で表記する．ページ単体からなるページセット $\{p\}$ の存在も許可される．また，ページセット s にページ p を追加して新たに定義したページセットを $s \cup \{p\}$ と表記する．

ページセットは，ページセットをユーザが閲覧したときにユーザに与える情報の関する特徴量とページセットがどのようなページから構成されるかについての特徴量を持つ．これらの特徴量を定義するために，検索結果集合 P における語の出現状況から語の関係を表す詳細グラフを作成する．この詳細グラフを用いて，ページセットの特徴量を定義し，ランキングに用いる．

3.2 詳細グラフ

3.2.1 詳細関係

語によって，概要的な内容を記述したページに出現しやすかったり，詳細を記述したものに出現しやすかったりするなどの傾向があり，これらの違いを表現する

表 1 B が A の詳細語であるかを判定するための 2x2 分割表
Table 1 Contingency table for query keyword A and candidate detailing keyword B.

	B を含む	B を含まない	合計
タイトル中に A を含む	$DF(intitle(A) \wedge B) \cdots (a)$	$(b)-(a)$	$DF(intitle(A)) \cdots (b)$
タイトル以外に A を含む	$(c)-(a)$	$(d)-(c)+(a)-(b)$	$(d)-(b)$
どこかに A を含む	$DF(A \wedge B) \cdots (c)$	$(d)-(c)$	$DF(A) \cdots (d)$

ために，検索結果集合での語の出現状況から語の詳細関係を定義する．検索結果集合にあまり出現しない語はそのトピックにおいてあまり重要でないと考え，検索結果集合 P に含まれる語のうち，DF の上位 l 位に含まれる語のみを詳細語の候補とし，詳細語候補集合 T とする．共起度を以下のように定義する．

$$cooc(t_i, t_j) = DF(t_i, t_j)/DF(t_i) \quad (2)$$

ただし，$DF(t_i, t_j)$ は語 t_i，t_j をともに含む文書数，$DF(t_i)$ は語 t_i を含む文書数である．ここで，語 t_j が語 t_i より詳細である $(t_i \prec t_j)$ 状態は，語 t_i，t_j が同時に出現する文書数が十分ある中で，語 t_j が含まれる文書中で，語 t_i が出現する確率が高く，その逆はいえない状態と定義する．さらに，詳細関係は推移すると定義する．この条件を式で表すと以下のようになる．

$$DF(t_i, t_j)/|P| > \theta_{DF} \quad (3)$$

$$cooc(t_j, t_i) > \theta_{cooc} \wedge cooc(t_i, t_j) < \theta_{cooc} \quad (4)$$

$$\forall t \prec t_j, t \prec t_i \quad (5)$$

また，定義より，以下の 2 式が成り立つ．

$$t_1 \prec t_2 \Rightarrow \forall t \prec t_1, t \prec t_2$$

$$t_2 \prec t_3 \Rightarrow \forall t \prec t_2, t \prec t_3 \quad (6)$$

上式より，$t_1 \prec t_3$ が得られるので，つまり，以下がつねに成り立つ．

$$t_1 \prec t_2 \wedge t_2 \prec t_3 \Rightarrow t_1 \prec t_3 \quad (7)$$

よって，詳細関係 $\prec$ には同じページ集合 P において推移性が成り立つ半順序関係である．

3.2.2 グラフの定義

この詳細関係を利用して，上位に概要的な語が位置し，枝をたどるにつれ，より詳細な語に至るようなグラフ，詳細グラフを定義する．各語をノードとし，以下が成り立つ場合に t_i から t_j へ有向枝を張るものとする．

$$t_i \prec t_j \wedge \nexists t, t_i \prec t, t \prec t_j \quad (8)$$

クエリ q によって得られた検索結果集合内では，どのページにも必ずクエリ q は含まれるので，このグラフのルートはクエリ q をキーワードに持つ．さらに以下が成り立つとき，t_i と t_j は密接な関係にあると考えられるので，ノードを集約し，t_i，t_j に対応するノードの親子関係を引き継ぐ．

$$cooc(t_j, t_i) > \theta_{cooc} \wedge cooc(t_i, t_j) > \theta_{cooc} \quad (9)$$

しかし，一般的な語 t_c が候補語集合に紛れ込んでいる場合，その語が詳細グラフの上位のノードと認識されるおそれがある．そのため，前述した Oyama らによる詳細語発見のアルゴリズムにより，t_c が詳細語の候補かどうかを判定し，詳細語でなければ除外する．問合せ回数削減のため，本稿では，ルートに直接つながっているノードが持つキーワードについてのみ詳細語かどうかを判定する．このようにしてクエリ “ハンガリー” に対して**図 3** のような DAG が得られる．

3.3 特 徴 量

ページセットの特徴を表す関数として，被覆度と重複度を以下のように定義する．ただし，いずれの関数の値域も $[0, 1]$ である．

3.3.1 被 覆 度

詳細グラフにおいて，共起関係から上位のノードに含まれるキーワードは内容を概要的に表す語，下位のノードは内容を詳細に表す語と考えることができる．下位のノードに含まれるキーワードについて詳細語か否かを検証しないのは，そのキーワードが一般的な語であった場合，出現数が多いと考えられるので，IDF による重みづけを行うことにより，影響を少なくできるからである．

ページセット s に語 t が含まれているかを表す関数として，$c(s, t)$ を以下のように定義する．ただし，$t \in s$ は s が t を含む，$t \notin s$ は s が t を含まない状態を示す．

$$c(s,t) = \begin{cases} 1, & t \in s \\ 0, & t \notin s \end{cases} \quad (10)$$

重みつき被覆度 cov_w を以下のように与える．

$$cov_w(s, g_q) = \frac{1}{|child(n_q)|} \sum_{n \in child(n_q)} cov_{node}(s, n) \quad (11)$$

$$cov_{node}(s, n) = \frac{\sum_{t \in g_n} IDF(t) c(s, t)}{\sum_{t \in T} IDF(t)} \quad (12)$$

g_q はクエリ q によって得られた詳細グラフである．n_q

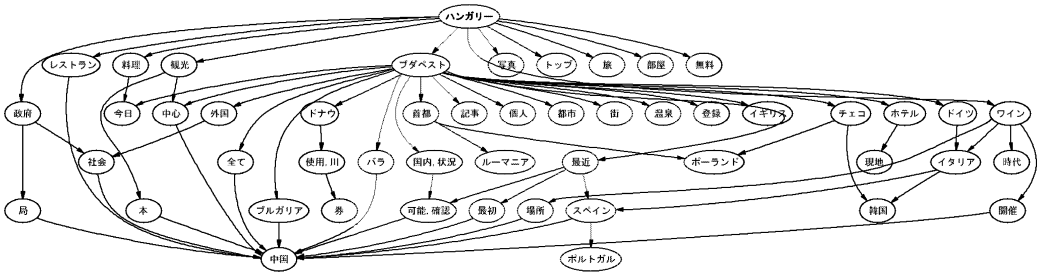

図 3 詳細グラフの例
Fig. 3 Example of detailing graph.

はクエリ q をキーワードに持つノードであるが，詳細グラフは検索結果集合から生成することを前提とするため，すべてのページはクエリとして与えたキーワード q を含むので，ルートは必ず q をキーワードに含む．つまり，n_q はここではルートである．$child(n_q)$ はノード n_q に直接つながっている子ノードの集合とする．たとえば，図 3 では，n_q は “ハンガリー” というラベルのついたノードであり，“ブダペスト” というラベルのついたノードは $child(n_q)$ に含まれるが，“首都” というラベルのついたノードは $child(n_q)$ に含まれない．また，$|child(n_q)|$ は $child(n_q)$ に含まれるノードの数，g_n はノード n をルートとする詳細グラフ g_q の部分グラフである．IDF は以下のように定義する．

$$IDF(t) = \log \frac{|P|}{DF(t)} + 1 \tag{13}$$

$DF(t)$ は P において，語 t を含む文書数である．

図 4 にクエリ q の検索結果についての詳細グラフの模式図を示す．図 4 で三角形で示されているのが，ノード n_q に直接つながったノードをルートとする部分グラフである（ただし，詳細グラフは DAG なので，部分グラフ間に重複が存在する場合もある）．各部分グラフはクエリ q の主要なサブトピックを表していると考えられる．cov_{node} は，ページセットが各部分グラフに含まれる語のどのくらいの語を含んでいるか，つまり対応するサブトピックの内容をどの程度含んでいるかを $[0,1]$ で表している．語の重みとして IDF を採用し，DF の小さいもの，つまり詳細な内容を説明するために用いられている語の重みを重くしている．cov_w はそれらの平均をとったものであり，cov_{node} の値域が $[0,1]$ なので，それぞれのサブトピックを同じ重みで扱っている．これによって全容検索の目的である，内容の広さと深さの両方を兼ねそなえたページセットを表現している．

3.3.2 重 複 度

重みつき重複度 dup_w を以下のように定義する．

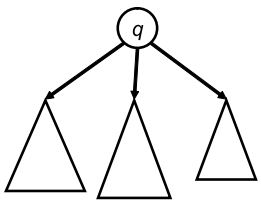

図 4 詳細グラフの模式図
Fig. 4 Image of detailing graph.

$$dup_w(s, g_q) = \frac{1}{|child(n_q)|} \sum_{n \in child(n_q)} dup_{node}(s, n) \tag{14}$$

$$dup_{node}(s, n) = \frac{\sum_{t \in g_n} IDF(t) ov(s, t)}{\sum_{t \in g_n} IDF(t)} \tag{15}$$

ただし，g_n は詳細グラフ g のノード n をルートとする部分グラフ，$ov(s,t)$ は，s 内の複数のページに語 t が含まれているときには 1，それ以外のときには 0 を返す関数であり，以下のように定義される．

$$ov(s,t) = \begin{cases} 1, & |\{p | p \in s, t \in p\}| \geq 2 \\ 0, & |\{p | p \in s, t \in p\}| < 2 \end{cases} \tag{16}$$

重みつき重複度についても重みつき被覆度と同様に，dup_{node} は，図 4 の部分グラフ，つまり，各サブトピックにおいて，複数のページに含まれる語がどのくらいあるかを表している．dup_{node} では，DF の小さいもの，つまり詳細グラフ内の下位に近いノードの重みを重くしている．これは DF の大きいものは概要について述べていると考えられるので，内容が異なる文書に出現する頻度が高いと考えられ，重みは低くするべきと考えられる．また，DF の小さいものは詳細について述べており，内容の異なる文書には出現する頻度が低く，このような語が重複して出現する場合は同じ内容を述べていると考えられるので，重みは高くするべきであるという考えに基づいている．また，dup_w は dup_{node} の平均をとったもので，dup_{node} の値域

が $[0,1]$ なので，dup_w でもそれぞれのサブトピックを同じ重みで扱っている．

3.4 解とすべきページセット

解とすべきページセット s は，ページセットが表す内容ができるだけ多くの必要な情報を含む，つまり $cov_w(s,g_q)$ がなるべく大きく，ページ間の内容的重複がなるべく少ない，つまり $dup_w(s,g_q)$ がなるべく小さいものである．

4. アルゴリズム

4.1 全体の流れ

全容検索の処理の流れは大きく分けて以下の 3 段階である．

- 検索結果集合の取得
- 詳細グラフの計算
- ランキングの計算

検索結果集合の取得では，既存のページごとの検索エンジンにより，与えられたキーワードについて検索を行い，ランキングの上位 l 件を取得し，P とする．他の 2 つについては以下に詳細を示す．

4.2 詳細グラフの計算

詳細グラフは以下のような手順で計算する．

(1) 取得したページ集合 P 内での共起度を計算し，DF の上位 k 語を詳細語候補集合 T とする．

(2) クエリに対応するノードのみからなる詳細グラフ g を作成する．

(3) $DF(t)$ の大きい順に，$t \in T$ のそれぞれに対応するノードを作成し，式 (3)，(4)，(5) の条件を検証し，子として g に追加できる最も深い位置に追加する．

(4) $n \in child(n_q)$ に対して，P 内での出現文書数の多いものから順に詳細語[12] の判定アルゴリズムによって詳細語かどうか判定し，詳細語でなかった場合は，以下のようにし，n とともにそれにつながっていた枝を削除する．

$$child(n_q) \leftarrow child(n_q) \cup child(n) \backslash \{n\}$$

更新した $child(n_q)$ に対してもすべてのノードを検証するまで繰り返す．

(5) 語 t_i，t_j が式 (9) を満たすとき，対応するノードをマージし，各ノードの親と子を引き継ぐ．

また，式 (4) の条件より以下が成り立つ．

$$t_i \prec t_j \Rightarrow DF(t_i) > DF(t_j) \tag{17}$$

このため，上記の手順の (3) では，t_j に対応するノードを追加する時点では t_j の子になるべきノードは g に存在していないため，親および祖先にあたるノードの条件のみ検証すればよい．

4.3 ランキングの計算

$cov_w(s,g_q)$ がなるべく大きく，$dup_w(s,g_q)$ がなるべく小さい s を発見し，ランキングする．普通に計算すると，$O(2^{|P|})$ かかり，求めるページセットを構成するページの最大値が m であっても $O(|P|^m)$ かかり計算量が非常に大きい．このため，cov_w および dup_w の性質を用いて検証するページセットの数を減らす必要がある．

まず，定義より，$ov(s \cup \{p\},t) \geq ov(s,t)$ なので，以下が成り立つ．

$$\begin{aligned} & dup_{node}(s \cup \{p\}, n) \\ &= \frac{\sum_{t \in g_n} IDF(t) ov(s \cup \{p\}, t)}{\sum_{t \in g_n} IDF(t)} \\ &\geq \frac{\sum_{t \in g_n} IDF(t) ov(s \cup \{p\}, t)}{\sum_{t \in g_n} IDF(t)} \\ &= dup_{node}(s,n) \end{aligned} \tag{18}$$

よって，$dup_w(s \cup \{p\}) \geq dup_w(s)$ が成り立つため，以下の式も成り立つ．

$$\begin{aligned} & dup_w(s,g_q) > \theta_{dup} \\ & \Rightarrow \forall p \in P, dup_w(s \cup \{p\}, g_q) > \theta_{dup} \end{aligned} \tag{19}$$

このため，$dup_w(s,g_q) > \theta_{dup}$ となる s を含む s' は検証の対象から外すことができる．

$c(s \cup \{p\},t) \geq c(s,t)$ より同様に以下が成り立つ．

$$cov_w(s \cup \{p\}) \geq cov_w(s,g_q) \tag{20}$$

$cov_w(s,g_q)$ の値はページセットにページを追加する順序によらないため，$cov_w(s \cup \{p\}, g_q) = cov_w(s,g_q)$ となった場合，$s \cup \{p\}$ は冗長であると考えられるため，$s \cup \{p\}$ を含むページセットは検証の対象から外すことができる．

また，検証対象を減らすために，任意の $s \in S_{i+1}$ は以下の条件を満たすとする．

$$cov_w(s,g_q) > \max_{s' \in S_i} cov_w(s', g_q) \tag{21}$$

この制約により，検証するページセットの数を大きく削減することができる．しかし，高い被覆度を持つページが存在する場合，それを超える被覆度を持つページセット s のうち，$|s| \geq 3$ を超えるものについては計算されない可能性があるという問題点がある．この場合，検索の解としては単一ページで十分であるとも考えられる．

S_i を $|s| = i$ となるページセットの集合，R をラン

表 2 全容検索と他の手法の比較（クエリ：“鳥インフルエンザ” の場合）
Table 2 Comparison between overview search and other method when query is “avian flu.”

検索手法	(a)	(b)	ID	(c)	(d)	(e)
全容検索	1.000	0.408	Page1-1	0.391	30	70
			Page1-2	0.825	7	75
			Page1-3	0.127	70	45
ページごとの検索（被覆度）	0.987	0.987	Page2-1	0.950	1	51
			Page2-2	0.882	2	78
			Page2-3	0.855	3	76
ページごとの検索（Google）	0.813	0.721	Page3-1	0.577	20	1
			Page3-2	0.251	41	2
			Page3-3	0.534	24	3

ページセットの (a) 重みつき被覆度，(b) 重みつき重複度，
ページの (c) 重みつき被覆度，(d) 重みつき被覆度の順位，
(e) Google での順位

表 3 全容検索と他の手法の比較に用いたページ（クエリ：“鳥インフルエンザ” の場合）
Table 3 Pages obtained by overview search and other method when query is “avian flu.”

ID	タイトル	説明
Page1-1	鳥インフルエンザについての Q & A を更新しました	一般消費者向け情報
Page1-2	埼玉県／高病原性鳥インフルエンザに関する情報について	一般消費者/飼育者向け情報
Page1-3	鳥インフルエンザは大流行するか―バイオ企業の動向	ニュース記事
Page2-1	鳥インフルエンザ＆新型インフルエンザ情報	簡単な説明とリンク集
Page2-2	宮城県/畜産課/鳥インフルエンザについて	一般消費者/飼育者向け情報
Page2-3	埼玉県食品安全企画室/鳥インフルエンザに関する対応について	一般消費者/飼育者向け情報
Page3-1	厚生労働省：鳥インフルエンザに関する情報関連情報	トップページ
Page3-2	国民の皆様へ（鳥インフルエンザについて）	一般消費者向け情報
Page3-3	国立感染症研究所感染症情報センター：鳥インフルエンザ	トップページ

キングの対象となるページセットの集合とすると，ランキングの計算の手順は以下のようになる．

(1) $R \leftarrow \phi$, $S_1 = \{\{p_1\}, \{p_2\}, \cdots, \{p_n\}\}$,$i = 1$ とする．

(2) $S_{i+1} \leftarrow \phi$ とする．

(3) $s \in S_i$ に対して，$s' = s \cup \{p\}$ を計算する．
(ただし，$p \in P$,$p \notin s$,$dup_w(s') < \theta_{dup}$)

$$cov_w(s') > \max_{s'' \in S_i} cov_w(s'')$$

を満たす場合，$S_{i+1} \leftarrow S_{i+1} \cup \{s'\}$ とし，満たさない場合は，$R \leftarrow R \cup s$ とする．

(4) $S_{i+1} \neq \phi$ ならば，$i \leftarrow i + 1$ として，(2) に戻る．

(5) $s \in R$ について，$cov_w(s, g_q)$ を第 1 のキーとして降順に，$dup_w(s, g_q)$ を第 2 のキーとして昇順にソートしたものが求めるランキングになる．

5. 実　　験

5.1 実験環境

検索エンジンには，Google[13] を用い，検索結果の上位 100 件を取得し，それを検索結果集合 P とした．その中での DF 値の上位 100 語を詳細語候補集合 T とした．パラメータは，$\theta_{cooc} = 0.8$，$\theta_{dup} = 0.5$，$\theta_{DF} = 0.2$ とした．また，$|s| \leq 3$ に限定して実験を行った．

5.2 全容検索の例

全容検索の評価を行うため，(1) 全容検索によって検索したページセットのランキング，(2) 検索結果を重みつき被覆度でリランキングし，上位 3 件をページセットと見なしたもの，(3) Google の検索結果をそのまま上位 3 件をページセットと見なしたものを比較する．これらのページセットはそれぞれ，(1) 提案した手法によって閲覧を行った場合，(2) 被覆度の高いページ順に閲覧を行った場合，(3) Google のランキング順に閲覧を行った場合において，ユーザが同じページ数を閲覧した際に，得る情報をページセットとして表現している．**表 2** に，クエリを “鳥インフルエンザ” としたときのページセットおよびそれを構成するページについて被覆度や重複度などを，**表 3** にそのページのタイトルと内容の簡単な説明を記す．ID は表 2 と表 3 でのページの対応関係を表す識別子である．表 3 を補足すると，(1) の場合では，Page1-1，Page1-2 は鳥イン

フルエンザの FAQ であったが，Page1-1 には主に消費者向けに詳しい説明があり，Page1-2 は消費者向けの情報だけではなく，飼育者向けの情報も含んでおり，お互いに異なる情報を含んでいた．また，Page1-3 は鳥インフルエンザに対する企業の取り組みを紹介しており，他の 2 つとは異なる情報が書かれていた．これに対して，(2) の場合では Page2-1 は鳥インフルエンザについての簡単な説明と数十ページへのリンクが張られていた．また，Page2-2，Page2-3 はともに一般消費者と飼育者に向けた情報が書かれていたが，消費者向けの情報については Page1-1 ほど詳しい内容を含んでいなかった．(3) の場合では，Page3-1，Page3-3 は鳥インフルエンザについてのサイトのトップページでそのページ自体にはあまり情報が含まれておらず，リンク先の個別ページに実際の内容が記述されていた．Page2-1，Page3-1，Page3-3 のようなページでは，リンク先まで閲覧することによって実際の内容が得られると考えられるが，リンク先のページ数がそれぞれ 10 ページ以上と多く，多くのページを閲覧しないと全容の理解は難しく，効率的な閲覧にとっては不利であるといえる．また，表 2 から (2) の場合では，完全に必要な情報が得られるわけではないうえに，ページ間の内容の重複が非常に多く，効率的な情報の収集はできず，(3) の場合では，必要な情報が得られず，ページ間に重複した情報も多いことが分かる．

5.3 全容検索の評価

前節で述べた傾向が他のクエリに対してもあてはまるかを調べるために，4 つのクエリ（鳥インフルエンザ，ハンガリー，風力発電，京都）に対して，比較を行い，平均をとったものが**表 4** である．この結果より，全容検索と比較して，被覆度を重視したページごとの検索では，被覆度は若干劣るものの同等程度の値が期待できるが，重複度がきわめて高いため，非効率な閲覧しかできないことがいえる．また，検索結果からのランキングを重視したページごとの検索では，重複度はきわめて低いが，被覆度も低く，必要な情報を得られていないといえる．このような点から全容検索は，内容の被覆度，重複度の両方の面でバランスがとれた検索手法であるといえる．

また，実行時間については**表 5** に 4 つのクエリに対して，Google の検索結果の上位 100 件はあらかじめダウンロードしてある状態で実行にかかった時間を示す．各ラベルの意味はそれぞれ，“詳細グラフ” は詳細グラフの計算にかかる時間，“ランキング” は詳細グラフが計算してある状態でページセットを生成し，ランキングの計算にかかった時間，“全体” はクエリを入力してからランキングが出力されるまでの時間を表し，いずれも単位は秒である．詳細グラフの計算については，10 秒以内で済む場合もあれば，1 分以上かかる場合もある．これはグラフの構造によって詳細語かどうかを判定すべき語が数が多いため時間がかかっていると考えられる．これを高速化するためには，一般的な語を集めて辞書をつくっておき，詳細語かどうかを判定する前にそれに照合することが考えられる．ページセットの計算については，どれも 1 分以上かかっており，クラスタリングなどを使い，検証するページセットの数を減らすなどして高速化する必要がある．

表 4 全容検索と他の手法との比較（平均値）
Table 4 Comparison between overview search and other method (average).

検索手法	(a)	(b)	(c)
全容検索	0.979	0.472	0.415
ページごとの検索（被覆度）	0.964	0.866	0.745
ページごとの検索（Google）	0.463	0.280	0.218

(a) ページセットの重みつき被覆度の平均
(b) ページセットの重みつき重複度の平均
(c) ページの重みつき被覆度の平均

表 5 全容検索の実行時間（単位：秒）
Table 5 Execution time of overview search.

クエリ	詳細グラフ	ランキング	全体
鳥インフルエンザ	17	74	146
ハンガリー	65	88	211
風力発電	9	86	138
京都	39	75	146

5.4 クラスタリングとの比較

クラスタリングサービスを使った場合との比較として，clusty [2] で鳥インフルエンザについて調べた場合について考える．1 階層目のクラスタとして，**表 6** のようなラベルのついたクラスタが得られる．各クラスタにはページが重複を許して分類されている．まず，知識のないユーザがこの結果から全容を理解しようとする場合，各クラスタごとにページを選んで閲覧することが考えられる．この場合，クラスタの数が 10 個あるので，10 ページを閲覧する必要がある．もちろん，複数のクラスタに分類されているページもあるので，そのようなページが複数のクラスタの内容を代表していると分かれば閲覧するページ数を削減することができる．しかし，clusy をはじめ，他のクラスタリングサービスでも，どのページが各クラスタを代表しているかの情報は提供されていない．また，各ページが所属するクラスタの数も多くはないため，閲覧ページ数の削減は難しい．これに対して 5.2 節で示した提案手法の例では 3 ページだけで全容を表現しており，

表 6 クラスタリングの例
Table 6 Example of clustering.

クラスタのラベル	クラスタ内のページ数
高病原性鳥インフルエンザ	66
鳥インフルエンザ対策	24
インフルエンザウイルス	18
安全，委員会	15
新聞	13
新型インフルエンザ	15
野鳥	11
ヒト	12
大量死	3
鳥インフルエンザ流行	4

効率的である．さらに各クラスタから代表するページを選択するとき場合，ユーザはタイトルとスニペットのみで判断することになり，特に予備知識のないユーザにとっては難しい．このように現在あるクラスタリングサービスのみでは予備知識のないユーザが全容を理解できるようなページセットを発見することは困難であり，提案手法の方が優れていると考えられる．

6. おわりに

本稿では，ページセットを検索の解とする全容検索を提案した．全容検索は，内容的な広さと深さを両立させ，ページ間の内容の重複の少ないページ集合を発見する．検索結果集合から詳細グラフを生成して，語の詳細関係を求め，それを用いてページセットを生成するアルゴリズムを示した．また，実験により，従来のページごとの検索に対する優位性が確認された．今後は，アルゴリズムの高速化，ページ間の関係の視覚化，リンク分析の導入などを検討していく予定である．

謝辞 本研究の一部は，文部科学省 21 世紀 COE 拠点形成プログラム「知識社会基盤構築のための情報学拠点形成」（リーダ：田中克己，平成 14–18 年度）および文部科学省研究委託事業「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」，異メディア・アーカイブの横断的検索・統合ソフトウェア開発（研究代表者：田中克己），文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」，計画研究「情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究」（研究代表者：田中克己，A01-00-02，課題番号：18049041）によるものです．ここに記して謝意を表します．

参考文献

1) 徳永健伸：情報検索と言語処理，東京大学出版会 (1999).
2) clusty.jp. http://clusty.jp/
3) Yumoto, T. and Tanaka, K.: Finding Pertinent Page-Pairs from Web Search Results, *Proc. 8th International Conference on Asian Digital Libraries* (*ICADL2005*), pp.301–310 (2005).
4) Yumoto, T. and Tanaka, K.: Page Sets as Web Search Answers, *Proc. 9th International Conference on Asian Digital Libraries* (*ICADL2006*) (2006).
5) Brin, S. and Page, L.: The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, Vol.30, No.1–7, pp.107–117 (1998).
6) Sun, J.-T., Wang, X., Shen, D., Zeng, H.-J. and Chen, Z.: CWS: A Comparative web Search System, *WWW '06: Proc. 15th international conference on World Wide Web*, New York, NY, USA, pp.467–476, ACM Press (2006).
7) Cutting, D.R., Pedersen, J.O., Karger, D. and Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, *Proc.15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.318–329 (1992).
8) Tajima, K., Mizuuchi, Y., Kitagawa, M. and Tanaka, K.: Cut as a Querying Unit for WWW, Netnews, and E-mail, *Proc. ACM Hypertext '98*, pp.235–244 (1998).
9) 風間一洋，原田昌紀，佐藤進也：サーチエンジンの検索結果のマルチレベル・グルーピングの評価，コンピュータソフトウェア，Vol.17, No.4, pp.354–365 (2000).
10) Toda, H., Kataoka, R. and Kitagawa, H.: Topic Structure Mining for Document Sets using Graph-Based Analysis, *Proc. 17th International Conference on Database and Expert Systems Applications* (*DEXA 2006*) (2006).
11) 尾暮拓也，中田圭一，古田一雄：コミュニティオントロジーを利用した情報検索，社会技術研究論文集，Vol.3, pp.102–110 (2005).
12) Oyama, S. and Tanaka, K.: Query Modification by Discovering Topics from Web Page Structures, *Proc. 6th Asia Pacific Web Conference* (*APWEB'04*), Lecture Notes in Computer Science, Vol.3007, pp.553–564 (2004).
13) Google. http://www.google.com/

(平成 18 年 9 月 15 日受付)
(平成 19 年 2 月 27 日採録)

(担当編集委員 石川 博，有次 正義，片山 薫，木俵 豊，中島 伸介)

湯本　高行（正会員）

兵庫県立大学大学院工学研究科電気系工学専攻助教．2007 年京都大学大学院情報学研究科社会情報学専攻博士後期課程修了．博士（情報学）．情報検索，情報統合の研究に従事．ACM，IEEE，日本データベース学会各会員．

田中　克己（正会員）

京都大学大学院情報学研究科社会情報学専攻教授．1976 年京都大学大学院修士課程修了．工学博士．主にデータベース，マルチメディアコンテンツ処理の研究に従事．IEEE Computer Society，ACM，人工知能学会，日本ソフトウェア科学会，日本データベース学会各会員．