新しい連結性概念と Web ページのグループ化への応用

A New Notion of Connectivity and its Application to Web Page Grouping

正田 備也♥ 高須 淳宏♦ 安達 淳♦

Tomonari MASADA Atsuhiro TAKASU
Jun ADACHI

本論文では，ハイパーリンクから得られる情報のみを用いて Web ページをグループ化する手法を提案する．グループ化のねらいは，Web 上での効果的な検索やマイニングの単位として利用できる，適度な大きさの Web ページの集合を構成することにある．提案手法は，強連結成分の細分化としてグループ化を行う．その際，グループの大きさは，閾値パラメータと呼ばれるパラメータを調節することで制御できる．また，本論文は 500 万の Web ページを対象とする実験の結果を含む．

This paper proposes a method for grouping Web pages based only on hyperlink information. The aim of grouping is to construct Web page sets of moderate sizes which can be utilized as units for effective search and mining on the Web. Our method provides groups as subdivisions of strongly connected components. Group sizes can be controlled by adjusting a parameter, called threshold parameter. This paper also includes experimental results of groupings of 5 million Web pages.

1. はじめに

1.1 研究の目的

本研究では，ハイパーリンクによるWebページの参照関係のみを用いてグループ化を実現する手法を提案する．グループ化のねらいは，効果的な検索やマイニングの単位として，適度な大きさのWebページの集合を構成することにある．

1.2 従来研究

従来の文書クラスタリングは，各文書に一定の次元のベクトルを割り当て，そのベクトルの類似性によって元の文書の類似性を評価する，という枠組みを専ら採用している．各文書に対応するベクトルは，単語の種類に匹敵する次元をもち，文書d_jに割り当てられるベクトルw_jの，単語t_iに対応するエントリ$w_{i,j}$は，t_iが文書d_jに出現する回数と，t_iが現われる文書の総数n_iとからtf・idfと呼ばれる式によって算出される．このように，ベクトルへ写像された文書の集合のクラスタリングには，Scatter/Gather[1]やCLARANS[2]を用いることができる．しかし，この枠組みを採用すると，形態素解析などによる特徴量抽出に要する時間が無視できない．さらに，文書集合をベクトル集合へ写像した後も，ベクトルの次元数が全文書に現われる単語の種類に匹敵するため，「次元の呪縛」と呼ばれる困難に見舞われる．特異値分解法[3]のような次元削減の手法が文献検索の分野で活用されるゆえんである．よって，文書のベクトル表現に基づくクラスタリングは，Webのような大規模な文書集合には必ずしも有効でない．

そこで，Webページがハイパーリンクによって参照しあっているという事実の利用が考えれられる．リンク構造は，Webページを頂点，リンクを枝とみなすことによって，一つのグラフとみなされる．よって，グラフの頂点のクラスタリング手法を適用できる．[4]では重要でないリンクを切断することで連結な部分グラフを取り出す手法が提案されている．しかし，リンクの重要性が上述のベクトルモデルに基づいて定量的に表現される．よって，リンク構造をグラフとして捉えている点は注目に値するものの，スケーラビリティの観点からWebには適用しがたい．

もちろん，リンク構造のグラフとしての側面から得られる情報だけを利用して頂点をグループ化することもできるが，グラフに関わる問題には困難なものが多い．しかし，グラフの隣接行列やラプラシアンの固有ベクトルを利用することで，カットを小さくするという意味で良いグループ化を得る近似アルゴリズムが提案されている[5][6]．ところがこれらは，隣接行列やラプラシアンが対称となる無向グラフを扱っており，Webのリンク構造には直接適用できない．そこで，リンク構造に対応する有向グラフの隣接行列AそのものではなくA^TAやAA^Tという対称化された行列に同趣旨の手法を適用し，Webページをグループ化する試みもある[7]．だが，この手法については有効性を疑問視する声がある[8]．

その一方，Webグラフの効果的なマイニングや可視化を目的とする研究の中には，個別のWebページよりも大きく，かつ，一つのサーバよりも小さなWebページの集合を構成する興味深い手法が見られる[9][10]．しかし，いずれもページを同じグループにまとめる基準が，URLやサーバ内のディレクトリ構造など，ハイパーリンクとは無関係な情報にも依拠しつつ，ヒューリスティックに定められている難点がある．

1.3 研究の特色

本研究は，以上の既存研究とは異なり，有向グラフに特有な概念である強連結成分分解を出発点とし，リンク情報のみに依拠したグループ化を提案する．確かに，強連結成分分解自体が頂点のグループ化ではある．また，強連結成分分解は少ない計算量で遂行できる[11]．しかし[12]が実験的に確かめているように，Webのリンク構造上で強連結成分分解を行うと，一つだけ巨大な成分が構成される．さらに[13]より，Webのリンク構造では，ほぼ確実に，ちょうど一つの巨大な強連結成分が存在し，かつ，そのサイズが$\Theta(n)$となる．nはWebページの総数である．このように巨大なグループを与えるグループ化手法は，そのグループが雑多なページを含むと予想されるため，Webグラフ上での検索やマイニングの単位の作成としてのグループ化には向かない．

そこで，本研究では，**強連結成分の細分化**としてのグループ化手法を提供する．まず，リンク情報のみに基づきWebページ間に距離概念を導入する．具体的には，一つのページから別の一つのページへの，向きのついた移行のしやすさの指標として**ドリフト**という概念を定義し，これに基づいて，二つのページ間の距離をあらわす概念である**相互リンク距離**を定義する．次に，これらの概念を利用し，以下のようなグ

♥ 学生会員 東京大学大学院情報理工学系研究科
masasda@nii.ac.jp
♦ 正会員 国立情報学研究所 {takasu,adachi}@nii.ac.jp

ループ化アルゴリズムを提案する．まず，任意に一つのページを選び，これを今から構成するグループの中心をなすページとする．そして，当のページからの相互リンク距離が所与のパラメータτ以下のページだけを一つのグループとしてまとめる．この操作は，全ページがいずれかのグループに属するまで繰り返される．値τは，グループ化のアルゴリズム全体に対するパラメータであり，この値を調節することで，得られるグループの粒度を制御できる．このパラメータτを**閾値パラメータ(threshold parameter)**と呼ぶ．さらに，τをある決まった値以上に設定すると，アルゴリズムは強連結成分分解を与える．つまり，本研究の提案するグループ化手法は，この意味で強連結成分分解の一般化とみなされうる．そこで，このアルゴリズムによって構成される連結成分を，**パラメータ化された連結成分** (parameterized connected component) と呼び，PCCと略記する．本研究の特色をまとめると以下のようになる．

- ハイパーリンクから得られる情報のみに基づいてWebページをグループ化する．
- グループ化の理論的枠組みとして，ドリフトおよび相互リンク距離という新しい概念を導入している．
- グループの粒度を，閾値パラメータと呼ばれるパラメータの値を調整することで制御できる．
- PCCは，強連結成分の一般化とみなすことができる．

2. 新しい概念

2.1 ドリフト (Drift)

Webページは，ハイパーリンクを介した参照関係によって巨大な有向グラフを形成している．この有向グラフ$G=(V,E)$の頂点集合VはWebページの集合であり，有向枝の集合Eはハイパーリンクの集合である．$|V|=n$，$|E|=m$とする．頂点の集合Vを，自然数の集合$\{1,\ldots,n\}$として表し，各頂点をそのインデックスと同一視する．有向枝（以下「枝」と略記）は順序つきの頂点対として表される．頂点uからvへ張られた枝は(u,v)と表される．枝には**重み**$w(u,v)$が与えられている．重みは1未満の正実数とする．直感的には，重みは互いに隣接する頂点uからvへの関連の度合いが高いほど大きくなるように定める．また1未満の正の実定数cについて$\overline{w(u,v)}=\log_c w(u,v)$と定義される値を枝$(u,v)$の**対数重み**と呼ぶ．$c$が1未満の正実数であることより，重みが大きいほど対数重みは小さくなる．$(u,v)\notin E$のときは$w(u,v)=0$および$\overline{w(u,v)}=\infty$とする．重み$w(u,v)$を第u行，第v列のエントリとするn行n列の行列Wを**重み行列**と呼ぶ．**歩道**とは順序つきの枝集合$\pi=((u_1,v_1),\ldots,(u_l,v_l))$で$v_1=u_2,\ldots,v_{l-1}=u_l$を満たすものをいう．頂点の重複がない歩道を**パス**と呼ぶ．**閉路**とは，始点と終点が一致する歩道である．歩道$\pi=((u_1,v_1),\ldots,(u_l,v_l))$の重み$w(\pi)$を$w(\pi)=w(u_1,v_1)\cdots\cdots w(u_l,v_l)$と定義する．また，歩道の対数重み$\overline{w(\pi)}$を$\overline{w(\pi)}=\log_c w(\pi)$と定める．このとき歩道の対数重みは，歩道を構成する枝の対数重みの総和となる．単に歩道の**長さ**と言うときは，その歩道を構成する枝の数を意味する．つまり歩道$\pi=((u_1,v_1),\ldots,(u_l,v_l))$の長さは$l$である．

ここで**ドリフト**という新しい概念を導入する．頂点uからvへのドリフト$Dr(u,v)$とは，あらかじめ定められた演算子$\oplus$にしたがって，uからvへのすべての歩道の重みを集計したものをいう．つまり，uからvへの歩道の集合を$\Pi(u,v)$とすると，uからvへのドリフトは$Dr(u,v)\equiv\bigoplus_{\pi\in\Pi(u,v)} w(\pi)$と定義される．また，頂点$u$から$v$への**対数ドリフト**$\overline{Dr(u,v)}$を$\overline{Dr(u,v)}=\log_c Dr(u,v)$と定義する．おおよそ，頂点$u$から$v$へのドリフトが大きいほど（対数ドリフトが小さいほど）uからvへの関連の度合いが高い，ということになる．

2.2 相互リンク距離 (Mutual-link distance)

頂点uとvの間の**相互リンク距離**は$ML(u,v)\equiv\overline{Dr(u,v)}+\overline{Dr(v,u)}$と定義される．頂点集合$S\subset V$の**直径**を，$S$に属する二つの頂点間の有限な相互リンク距離の最大値と定める．ここで，具体的なドリフトの決め方，つまりは相互リンク距離の決め方を三つ示す．

(1) 演算子$\oplus$を，総和をとる操作と定める．このとき$Dr(u,v)$はn行n列の行列$\sum_{k=1}^{\infty}W^k$の第u行，第v列のエントリに一致する．よって，任意の頂点間の相互リンク距離を求めるには$\sum_{k=1}^{\infty}W^k=(I-W)^{-1}-I$を計算すればよい．ただし，枝の重みはこの和が収束するように正規化しておく．このように，重み行列の冪の和によって，ある頂点から別の頂点への関連性の度合いを表現する試みは過去にある[14]．だが，今回はスケーラビリティの観点から，この定義を採用しない．なぜなら，逆行列の算出というコストの大きな処理が必要だからである．しかし，[15]のように大規模な数値計算によってリンク解析を実践した例もあるため，この方法が常に非現実的なわけではない．

(2) 枝の重みをすべてcとし，$\oplus$を最大値をとる操作とする．このとき，すべての枝の対数重みが1となり，$\overline{Dr(u,v)}$はuからvへの歩道のうち最も短いものの長さに一致する．したがって，$ML(u,v)$はuからvへの最短パス長とvからuへの最短パス長との和となる．しかし，実際のWebグラフ上で，様々な頂点対についてこの意味での相互リンク距離を求めると，多くのページが密集していると分かる．このため，きめの細かいグループ化を実現し難く，今回はこの設定も採用しない．

(3) 枝(u,v)の重みを$w(u,v)=c^{d_u^+}$とする．d_u^+は頂点uの出次数である．つまり，枝の対数重みがその枝の始点の出次数に一致するように重みを決める．そして，$\oplus$は最大値をとる操作とする．このとき，$\overline{Dr(u,v)}$は，uからvへの歩道のうちそれに沿って存在する頂点の出次数の和が最小のものの，その最小値に一致する．したがって，$ML(u,v)$は，uとvを含む閉路のうち，それに沿って存在する頂点の出次数の和が最小のものの，その最小値に一致する．今回は，この設定を採用している．この設定の下では，出次数の大きいWebページが同じグループに属しにくくなる．よって，インデックス的なページやリンク集的なページなどリンク構造上重要なページが違うグループに属しやすい，という効果を期待できる．

なお，(1)から(3)いずれの設定を用いても，相互リンク距離は三角不等式を満たすことが容易に示せる．

2.3 パラメータ化された連結成分 (PCC)

パラメータ化された連結成分とは，**閾値パラメータ**と呼ばれるパラメータτの値に応じて，次の条件を満たすように構成された頂点の集合$S\subset V$のことをいう．つまり，$S\subset V$について**中心頂点**と呼ばれる頂点$u\in S$が存在し，閾値パラメ

ータ τ に対して，$ML(u,v) \le \tau$ がすべての $v \in S$ について成立するとき，このような S をPCCと呼ぶ．なお，2.2節のどの設定のもとでも，τ を一定の値以上にすれば，極大なPCCによる頂点集合 V のグループ化は強連結成分分解に一致する．このように，PCCへの分解は，強連結成分分解の一般化とみなすことができる．

また，本研究の提案するグループ化手法によれば，いずれのグループも一つの頂点からの相互リンク距離が τ 以下となるように構成される．したがって，この構成法と，相互リンク距離が三角不等式を満たすことから，グループの直径の上界は 2τ となる．

3. アルゴリズム

本研究の提案するグループ化アルゴリズムは下記のとおりである．このアルゴリズムは，各頂点の属する PCC の中心頂点を記録することによって，グループ化の情報を保持する．

1.すべての頂点がマークされていない状態にする．

2.マークされていない頂点から任意に頂点 u を選び出し u 自身を中心頂点とする PCC の構成員としてマークする．

3.u を含む閉路を列挙する．ただし，閉路上に存在する全頂点の出次数の和が τ 以下である範囲内で列挙する．

4.列挙された全閉路上の全頂点を u と同じグループに属するものとしてマークする．

5.すべての頂点が，いずれかの PCC の構成員としてマークされるまで，上の 2 から 4 を繰り返す．

上記のアルゴリズムとその実装方法についていくつかコメントを掲げる．

a) 今回の実装では，リンク構造に次の前処理を加えている．まず，入次数ゼロの頂点について，すべてのリンク先からのリンクを追加する．次に，出次数ゼロの頂点について，すべてのリンク元へのリンクを追加する．なお，出次数も入次数もゼロの頂点については，単独でグループをなすとみなす．

b) ステップ 2 での中心頂点の選び方は任意であるが，今回の実装では出次数の多い順に選んでいる．選ぶ順を変えれば得られる PCC も変わる．つまり PCC への分解は一意でない．

c) 閉路の列挙は，SSSP (single source shortest path) 問題を解くための Dijkstra の探索法[16]に fibonacci heap[17]を併用することで実現している．具体的には，ステップ 2 で選ばれた頂点 u から，まずはリンクを順方向にたどり探索パス上の終端をのぞく頂点の出次数の和が τ 以下の範囲で近い順に頂点を列挙する．次に，リンクを逆方向にたどり，探索パス上の始点をのぞく頂点の出次数の和が τ 以下の範囲で近い順に頂点を列挙する．最後に，両方の探索で列挙された頂点のうち，u からの相互リンク距離が τ 以下のものだけを，u を中心頂点とする PCC の構成員とする．なお，以上より，PCC への分解は APSP(all-pairs shortest paths)問題よりも難しくない．つまり[17]より時間計算量は $O(n^2 \log n)$ である．しかし，PCC の構成において探索は次数の和が τ 以下の範囲にしか及ばないため，この計算量はあくまで上界である．

d) 本アルゴリズムは並列化できる．なぜなら，異なる頂点からの Dijkstra の探索は独立に実行できるからである．ただし，異なる実行インスタンスによって同じ頂点が別々の PCC に属するとされた場合は，後処理によって一つの PCC のみに属するように定める．今回の実装では，相互リンク距離が最短の中心頂点をもつ PCC へ含ませている．

e) より少ないメモリで，より多くのリンク情報を格納できるようにするために，頂点のインデックスを利用し，リンク情報を二通りに場合分けして格納している[18]．一つは，リンク先の頂点のインデックスをそのまま保存する場合．もう一つは，リンク先の頂点のインデックスを，リンク元の頂点のインデックスとの差分で保存する場合．そして，インデックスの差が 1 バイトの場合のみ，後者の仕方で保存している．

4. 実験

今回の実験は，クローリングによって集められた 5,000,000 の Web ページを対象としている．実験環境は，Intel Xeon 搭載の Solaris マシン 8 台からなる．並列化にともない，異なる実行インスタンスが通信するために，マルチキャストを用いている．もちろん，単に頂点集合を等分して各インスタンスに分配し，各々与えられた頂点集合だけから中心頂点を選び探索をすれば，通信なしに並列化できる．しかし，頂点集合を等分しても，全処理がほぼ同時に終わるわけではない．実際[13]に示されているように，Web のリンク構造のような有向グラフでは，一つの頂点から枝をたどって到達できる頂点については，その個数が $\Theta(n)$ に達する場合がある一方，高々 $O(\log n)$ にとどまる場合もある．したがって，探索の及ぶ範囲は，どの頂点から出発するかによって大きく違う．そこで，別のインスタンスがやり残している仕事を手伝うという効果を得るために，マルチキャストを利用している．図 1 は，この実験で得た閾値パラメータ τ が 100 および 200 の場合の PCC のサイズの分布である．

今回はさらに別の実験によって，Web ページの総数および閾値パラメータ τ の変化に伴う実行時間の相対的な変化を調べている．上記の設定とは異なり，500,000 ページおよび 1,000,000 ページについて 1 台の Sun Blade 1000 上で実験を行っている（図 2）．τ が増加するほど時間が減少するのは，アルゴリズムの振る舞いが強連結成分分解に近づくからである．τ が小さいうちは，ページ数 2 倍の増加に対し実行時間が 4 倍の増加となっているが，大きな τ については実行時間の増加がほぼ 2 倍へと縮小している．

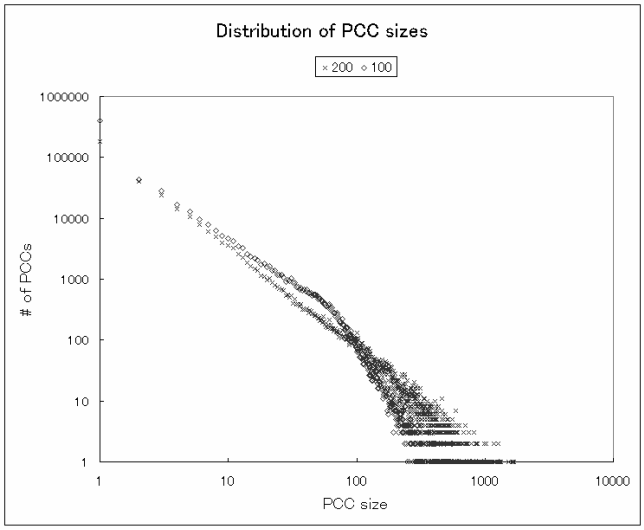

図 1 ：PCC のサイズの分布

横軸が PCC のサイズ，縦軸が PCC の個数．閾値パラメータが 100 の場合のほうが，小さい PCC が多く，大きい PCC が少ない．

5. おわりに

今後は，まずアルゴリズムの高速化を追求する．Dijkstra

の探索に，より効果的なヒープを組み合わせるなど，すでに実験中である．また，並行して提案のグループ化手法の実際的評価を進める．具体的には，[19]で提案されているテキスト・ベースの文書検索の手法と組み合わせて，Web 検索の性能への寄与を調べる．さらには，ネットサーフィンのナビゲーションへの応用も検討している．Web ページのグループ化は，ハイパーリンクをグループ内リンクとグループ間リンクに区別する操作とも見なしうる．よって，提案手法の与えるこの区別を，ナビゲーション情報としてネットサーファーに提示することで，例えば，少ないクリック数でより多様なページを閲覧できるという網羅性の意味で優れたネットサーフィンを実現できるものになっているかなど，検証を行う予定である．

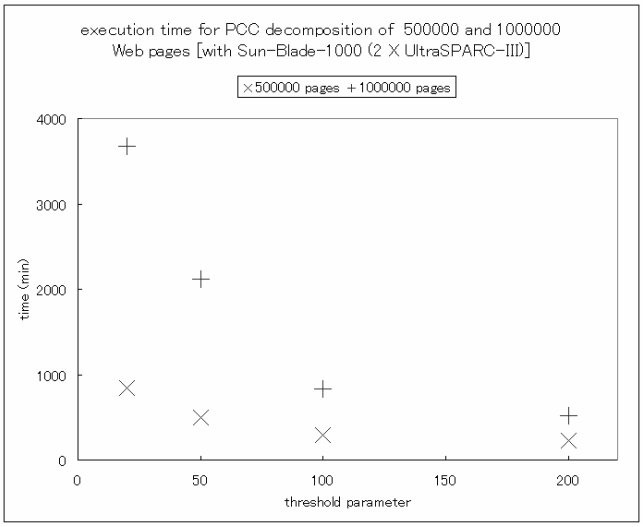

図 2：Web ページの数と閾値パラメータによる実行時間の変化
横軸が閾値パラメータ．縦軸は分で示した実行時間．
＋が 1,000,000 ページの場合．×が 500,000 ページの場合．

[謝辞]

今回の計算機実験は，国立情報学研究所の大山敬三教授のご協力がなければ，実現できませんでした．なお，本研究は文部科学省科学研究費補助金特定領域研究「情報学」の助成のもとに行われています．

[文献]

[1] D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey: "Scatter/Gather: A cluster-based approach to browsing large document collections", Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.318-329 (1992).
[2] R. T. Ng and J. Han: "CLARANS: A method for clustering objects for spatial data mining", IEEE Trans. on Knowledge and Data Engineering, Vol.14, No.5, pp. 1003-1016 (2002).
[3] M. W. Berry, S. T. Dumais, and G. W. O'Brien: "Using linear algebra for intelligent information retrieval", Dept. of Computer Science, Univ. of Tennessee, UT-CS-94-270 (1994).
[4] K. Tajima, Y. Mizuuchi, M. Kitagawa, and K. Tanaka: Cut as a querying unit for WWW, Netnews, and E-mail, Proc. of ACM Hypertext '98, pp. 235-244 (1998).
[5] L. J. Schulman: "Clustering for edge-cost minimization", Electronic Colloquium on Computational Complexity, Vol.6, No.035 (1999).
[6] R. Kannan, S. Vempala, and A. Vetta: "On clusterings - Good, bad and spectral", Proc. of the 41st Annual Symposium on Foundations of Computer Science, pp. 367-377 (2000).
[7] J. M. Kleinberg: "Authoritative sources in a hyperlinked environment", JACM, Vol.46, No.5, pp. 604-632, (1999).
[8] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas: "Finding authorities and hubs from link structures on the World Wide Web", Proc. of the 10th International World Wide Web Conference, pp. 415-429, (2001).
[9] L. Terveen, W. Hill, and B. Amento: "Constructing, organizing, and visualizing collections of topically related Web resources", ACM Trans. on Computer-Human Interaction, Vol.6, No.1, pp. 67-94, (1999).
[10] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa: "Applying the site information to the information retrieval from the Web", WISE 2002, pp. 83-92 (2002).
[11] E. Nuutila and E. Soisalon-Soininen: "On finding the strongly connected components in a directed graph", Information Processing Letters, Vol.49, No.1, pp. 9-14, (1994).
[12] A. Z. Broder, S. R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener: "Graph structure in the Web", Proc. of the 9th International World Wide Web Conference, pp. 309-320 (2000).
[13] C. Cooper and A. Frieze: "The size of the largest strongly connected component of a random digraph with a given degree sequence", preprint. available at http://www.math.cmu.edu/~af1p/papers.html
[14] L. Katz: "A new status index derived from sociometric analysis", Psychometrika, pp. 39-43 (1953).
[15] 安村賢英，川原稔，岩下武史，金澤正憲: "Web コミュニティ発見のための大規模有向グラフに対するデータ圧縮計算手法の VPP への実装", 京都大学大型計算機センター研究開発部 研究発表報告集, 第 17 号, pp. 71-80 (2002).
[16] E. W. Dijkstra: "A note on two problems in connexion with graphs", Numer. Math., Vol. 1, pp. 269-271 (1959).
[17] M. L. Fredman and R. E. Tarjan: "Fibonacci heaps and their uses in improved network optimization algorithms", Journal of the ACM, Vol.34, No.3, pp.596-615 (1987).
[18] K. Randall, R. Stata, R. Wickremesinghe, and J. Wiener: "The link database: Fast access to graphs of the Web", Research Report 175, Compaq Systems Research Center, Palo Alto, CA, 2001. 15 (2001).
[19] T. Kanazawa, A. Aizawa, A. Takasu, and J. Adachi: "The effects of the relevance-based superimposition model in cross-language information retrieval", Lecture Notes in Computer Science 2163, pp.312-324 (2001).

正田 備也 Tomonari MASADA
東京大学大学院情報理工学系研究科博士課程在学中．1995 年東京大学大学院理学系研究科情報科学専攻修士課程修了．情報検索に関連する研究に従事．情報処理学会学生会員．

高須 淳宏 Atsuhiro TAKASU
国立情報学研究所/総合研究大学院大学助教授．データベースシステム，機械学習の研究に従事．電子情報通信学会，情報処理学会，人工知能学会，ACM，IEEE 各会員．

安達 淳 Jun ADACHI
1981 年東京大学大学院工学系研究科博士課程修了．工学博士．東京大学大型計算機センター，文部省学術情報センターを経て現在国立情報学研究所教授．東京大学大学院情報理工学系研究科教授を併任．データベースシステム，分散処理システム，情報検索，電子図書館システム等の開発研究に従事．電子情報通信学会，情報処理学会，IEEE，ACM 各会員．