情報処理学会論文誌　データベース　Vol. 4　No. 2　142–157 (July 2011)

情報伝搬を考慮したグラフ分析による Twitter ユーザランキング手法

山口祐人[†1]　天笠俊之[†1,†2]
髙橋翼[†1,*1]　北川博之[†1,†2]

近年，Twitter と呼ばれるマイクロブログサービスが爆発的に普及している．多様な情報が時々刻々と発信される Twitter は，新しい情報源として注目を集めている．Twitter 上には様々なユーザが存在し，それぞれが自らの興味や嗜好に基づいた情報発信を行っている．なかには有用な情報を多く発信し，他のユーザへ大きな影響を与えるようなユーザも存在する．そのようなユーザの発見は，有用な情報の発見やマーケティングなどの様々な目的に必要とされ，さかんに研究されている．Twitter では，有用な情報はリツイートと呼ばれる他の情報を引用する機能によってユーザ間を広く伝搬していく．よって，より広く伝搬する情報を発信するユーザは有用である可能性が高い．しかし，従来のユーザランキング手法は，ユーザ間の関係を表すソーシャルグラフのみを解析しており，リツイートを考慮していない．本研究では，ソーシャルグラフにリツイートによる情報伝搬を取り入れた User-Tweet Graph に，PageRank を拡張したリンク構造解析手法である ObjectRank を適用し，ユーザの評価を行う手法 TURank を提案する．また，他の手法との比較実験を通して，提案手法の有効性を示す．

Ranking Twitter Users Based on Information Propagation Graph Analysis

Yuto Yamaguchi,[†1] Toshiyuki Amagasa,[†1,†2]
Tsubasa Takahashi[†1,*1] and Hiroyuki Kitagawa[†1,†2]

Recently, a micro-blogging service called Twitter has grown popular. It attracts a lot of attentions as a new type of information source, because diverse information is transmitted in real-time. There are a huge variety of Twitter users, and they transmit information based on their interests or preferences. Some users transmit a lot of useful information and have a great influence on other users. Therefore, identifying such users is considered a major research issue, because it is needed to identify useful information, to conduct marketing, and so on. Useful information spreads among users widely by Retweet which is a functionality of Twitter to cite other user's message. For this reason, users whose messages are frequently Retweeted are considered to be useful. However, conventional approaches only deal with social graphs consisting of relationships among users, and do not consider Retweet. In this paper, we introduce the User-Tweet Graph which incorporates information spread by Retweet into the social graph. Using such a graph, we also propose a user evaluation method called TURank. TURank analyzes the User-Tweet Graph using the concept of ObjectRank, which is a link analysis method extending PageRank. Experimental results show the effectiveness of the proposed approach.

1. 序　　論

近年のブログや SNS（Social Networking Service）などの普及により，Web 上で人々が情報を発信する機会が増加している．これらのサービスを利用するユーザは，自らの興味や嗜好に基づいた情報発信を行っている．そのため，有用な情報を発信するユーザを発見することができれば，価値ある情報の継続的獲得が可能であると考えられる．

一方，マイクロブログと呼ばれる，ブログと SNS の性質をあわせ持つサービスが普及している．マイクロブログの主な特徴として，投稿できるメッセージの長さに制限があることがあげられる．その制限によってユーザがメッセージを投稿する敷居が下がり，有用なものからそうでないものまで，様々な情報が時々刻々と発信されている．また，マイクロブログは SNS のようにユーザ同士がコミュニケーションをとることができるという特徴も持っている．

マイクロブログの中でも，Twitter[1)] が特に普及している．Twitter 上では，膨大な数のユーザがそれぞれ多様な情報をリアルタイムに発信している．そのため，Twitter は新しい情報源として多くの注目を集めている．Twitter を利用するユーザは，ツイートと呼ばれるメッセージを投稿することで情報を発信する．ツイートを投稿することをポストと呼ぶ．また，ユーザはフォローという機能を用いて欲しい情報を発信する他のユーザを登録し，そのユーザが投稿したツイートを継続的に受け取り，閲覧することができる．あるユーザをフォローするユーザをフォロワと呼ぶ．一般に，有用な情報を多く発信するユーザは，多くの

†1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†2 筑波大学計算科学研究センター
Center for Computational Sciences, University of Tsukuba
*1 現在，日本電気株式会社サービスプラットフォーム研究所
Presently with Service Platforms Research Laboratories, NEC Corporation

フォロワを集める傾向にある．

情報発信，消費という観点から，Twitter ではユーザがきわめて重要な役割を果たす．ユーザは自らの興味や嗜好に基づいて情報を発信する．有用な情報を多く発信するユーザもいれば，ただ情報を収集するのみで情報発信をしないユーザ，他のユーザとのコミュニケーションツールとして Twitter を利用するユーザもいる．すなわち，ユーザが発信する情報はユーザの興味や嗜好によって大きく異なり，Twitter の利用目的も様々である．そのため，多くのユーザにとって有用な情報を発信し，他のユーザへ大きな影響を与えるようなユーザの発見は，有用な情報の獲得やマーケティングなどの様々な目的に必要とされている．

近年，Twitter ユーザの評価，ランキングをする研究がさかんに行われている．Java ら[8] や Weng ら[16] は，フォローをユーザからユーザへ正の評価を与える投票であると見なし，その投票を用いてユーザの評価を行っている．これらの手法は，代表的なリンク構造解析手法の 1 つである PageRank[15] の手法を用いて，ユーザ同士のフォロー関係で構築されるソーシャルグラフを解析している．しかし，フォローによる投票は，ユーザが有用な情報を発信するかどうかを正しく評価できているとはいえない．Twitter 上では，ユーザは他のユーザからフォローされたとき，そのユーザが必ずしも有用な情報を発信するユーザでない場合でもフォローし返すという慣習がある．これは明らかに投票と見なすことはできない．この慣習を裏付けるものとして，全体の 72.4%ものユーザが自らのフォロワのうちの 80%をフォローし返していることが報告されている[16]．また，ツイートが有用であるからという理由でユーザをフォローするだけでなく，単に著名人だから，知人だからという理由でフォローすることも少なくない．これは，ユーザアカウント自体の性質への評価であり，ユーザが発信する情報への評価であるとはいえない．さらに，ある期間に有用な情報を発信していたユーザが，別の期間にも有用な情報を発信するとは限らない．しかし，ユーザは 1 度フォローをすると，フォローを解除することはめったにない．特に，多くのユーザをフォローするユーザは日々大量のツイートを受け取っているため，その中の 1 人のユーザが情報を発信しなくなってもそれに気づかない可能性が高い．すなわち，フォローはユーザアカウント自体の性質をある程度評価できているとはいえるものの，ある特定の期間にユーザが有用な情報を発信しているかどうかを正しく評価できているとはいえない．

Twitter 上では，リツイートによってユーザの間をツイートが伝搬する．リツイートとは他のツイートを自らのツイートへ引用し，再発信する機能である．ユーザはツイートをリツイートすることにより，フォロワにそのツイートを伝えることができる．Boyd ら[5] によると，ユーザがリツイートする目的は様々だが，主に有用であると判断したツイートをフォロワに伝えるために行われている．有用なツイートはリツイートによって次から次へと再発信され，ユーザの間を広く伝搬する．よって，リツイートはユーザからツイートへ正の評価を与える投票であるととらえることができる．リツイートによる投票を用いてユーザの評価，ランキングを行う研究も多く行われている[6),11),12),18)]．これらの研究は，ユーザが投稿したツイートがリツイートされた数を数え，ユーザのランキングを行っている．フォローとは異なり，リツイートはツイートへの評価であるため，ユーザが有用な情報を発信しているかどうかをとらえることができる．リツイートを用いたユーザのランキングは，フォローを用いたユーザのランキングとは大きく異なることが確認されている[6),11),18)]．しかし，いくつかのツイートが一時的に大量のリツイートを得たユーザは高い評価を得るが，そのユーザの他の大部分のツイートが有用とはいえないものであっても，それを考慮することはできない．また，リツイートされた回数を数えるだけでは，ツイートがどのようなユーザにリツイートされ，ユーザの間をどのように伝搬したかをとらえることはできない．

本稿では，先行研究[17] で提案した，フォローによるユーザアカウント自体への評価と，リツイートによるユーザのツイートへの評価の両方を考慮した Twitter ユーザの評価手法 TURank（Twitter User Rank）に詳細な定義を与える．TURank はリツイートによるツイートの伝搬をソーシャルグラフに取り入れた User-Tweet Graph を用いる．User-Tweet Graph はユーザとツイートをノードで表現し，フォロー，ポスト，リツイートをエッジで表現する．リツイートによるツイートの伝搬をモデル化し，グラフに取り入れることで，ツイートがどのようなユーザにリツイートされ，どのように伝搬したかを表現することができる．User-Tweet Graph に対して PageRank を拡張したリンク構造解析である ObjectRank[4] を適用し，ユーザの評価を行う．

本稿の以降の構成は以下のとおりである．まず 2 章では予備知識として，Twitter とリンク構造解析手法について説明する．3 章では提案手法の詳細について述べ，4 章では提案手法の有効性を示すために実施した評価実験について述べる．5 章では本研究に関連するいくつかの研究について概観し，6 章で本稿の結論を述べる．

2. 予 備 知 識

本章では，まず 2.1 節で Twitter について概観し，2.2 節では提案手法に用いられるリンク構造解析手法について説明する．

2.1 Twitter

Twitter[1] は，マイクロブログサービスの中でも最も注目を集め，爆発的に普及している

サービスである．2006 年のサービス開始から，2009 年末までに約 7,500 万人ものユーザを獲得している[14]．Twitter が有するユーザは，数が膨大であるだけではなく，その性質も多種多様である．世界各国からのユーザが存在するのはもちろん，近年では企業や政治家などの著名人，主要なニュースサイトまでもが Twitter の利用を開始し，情報を発信している．このように，Twitter からは様々な情報が発信されているため，多岐にわたるトピックに関する情報を得ることができる．

Twitter では，投稿できるメッセージの文字数が 140 文字以内に制限されている．メッセージをツイートと呼び，ツイートを投稿することをポストと呼ぶ．ポストされたツイートはそれぞれが個別の URL（パーマリンク）を持つため，Twitter アカウントを持たない人々でも自由に閲覧することが可能である．また，各ユーザのプロフィールページもパーマリンクを持つ．プロフィールページにはそのユーザがポストしたツイートが表示されているため，ある特定のユーザがポストしたツイートを網羅的に閲覧することも可能である．ただし，ツイートを非公開にしているユーザも存在する．それらのツイートは許可されたユーザしか閲覧することはできない．

Twitter にはフォローという機能がある．フォローとは他のユーザを登録し，そのユーザのツイートを “購読” する機能である．あるユーザをフォローしているユーザは，そのユーザのフォロワであるという．ユーザがポストしたツイートは，そのユーザのすべてのフォロワのタイムライン[⋆1]に即時に表示される．一般に，ユーザは自分が知りたい情報を発信するユーザや，著名人，知人など，ある一定の価値を見い出したユーザをフォローする傾向にある．すなわち，多くのユーザに有用であると見なされたユーザは多くのフォロワを集める．

他の大部分の SNS がユーザ同士の関係として双方向の友人関係を採用しているのに対して，Twitter はユーザ同士の関係として 1 方向の購読関係を採用している．これにより，Twitter ユーザは他のユーザの許可を必要とすることなく，誰でも自由にフォローすることができる[⋆2]．

リツイートは他のユーザのツイートを自分のアカウントで再発信する機能である．これにより，興味のある，または有用であるツイートを自分のフォロワにも伝えることができる．リツイートはもともとユーザの慣習から生まれた機能である．Twitter が正式にリツイート機能を提供するまでは，ユーザはリツイートしたいツイートの内容と，そのツイートをポストしたユーザ名を自らのツイートの中に引用し，必要であればさらにテキストを付加してポストしていた．これを非公式リツイートと呼ぶのに対して，Twitter が提供するリツイート機能を公式リツイートと呼ぶ．非公式リツイートとは異なり，公式リツイートは専用のリツイートボタンを押すだけでリツイートしたいツイートを自らのアカウントで再発信することできる．しかし，テキストの変更や追加はできない．

リツイートによって伝わったツイートをさらにリツイートするというように，リツイートは連鎖的に行われる．リツイートの連鎖により，多くのユーザに有用であると見なされるようなツイートは，ユーザの間を広く伝搬する．Kwak ら[11] は，ユーザが持つフォロワ数と，ツイートがリツイートによってどれだけのユーザに伝搬するかの間には相関がないことを示している．すなわち，フォロワ数の多寡にかかわらず，有用な情報はリツイートの連鎖によって多くのユーザへ伝搬する．これは，ユーザのフォロー関係のみを示すソーシャルグラフを解析するだけでは，有用な情報を発信するユーザの発見は困難であることを示している．

2.2　リンク構造解析

本節では，提案手法に用いられるリンク構造解析手法について説明する．2.2.1 項では，代表的なリンク構造解析手法である PageRank について説明し，2.2.2 項では，PageRank を拡張して考案された ObjectRank について説明する．

2.2.1　PageRank

PageRank[15] は，Web ページの重要性を評価するためのアルゴリズムである．PageRank は引用に基づく学術論文の評価に類似した，以下の発想をもとにしている．

- 多くの重要な Web ページからのリンクを得ているページは，やはり重要なページである．
- 乱発されたリンクにはあまり価値がない．

PageRank はランダムサーファモデルというモデルを用いている．ここで，Web ページを閲覧することを目的に Web 上を移動するランダムサーファを考える．ランダムサーファは，多くの場合リンクをたどって次のページへ移動するが，稀にリンクを無視してまったく無関係なページへ移動する場合がある．この，リンクと無関係な移動をランダムジャンプと呼ぶ．

PageRank による Web ページの評価値の計算は，Web ページの膨大なリンク構造を表すグラフ $G = (V, E)$ を対象に行われる．$V = \{v_1, \cdots, v_n\}$ をすべてのノード（Web ページ）の集合，E をすべてのエッジ（リンク）の集合とする．ランダムサーファがノード v_i から出発し，確率 d でリンクをたどって次のノードへ移動，または確率 $1 - d$ でランダムジャン

⋆1 タイムラインとは，ユーザ自らのツイートと，そのユーザがフォローするすべてのユーザのツイートを時系列順に表示させたものであり，各々のユーザに対して 1 つずつ存在する．
⋆2 非公開ユーザをフォローするには許可が必要である．

プする．これを 1 ステップとし，十分な時間が経過するまでこのステップを繰り返す．ノード v_i の重要度を表す評価値は，ある時間にランダムサーファがノード v_i にいる確率 r_i で与えられる．このとき，すべてのノードの評価値ベクトル $\mathbf{r} = [r_1, \cdots, r_n]^T$ は以下の式で表される．

$$\mathbf{r} = d\mathbf{A}\mathbf{r} + \frac{(1-d)}{|V|}\mathbf{u} \tag{1}$$

ただし，$\mathbf{A}$ は n 次正方行列であり，v_j から v_i へのエッジ (v_j, v_i) が存在するとき $a_{ij} = 1/OutDeg(v_j)$，存在しないとき 0 を要素として持つ．ここで，$OutDeg(v_j)$ はノード v_j の出次数である．また，$\mathbf{u} = [1, \cdots, 1]^T$ である．右辺第 1 項はリンクをたどって各ノードへ移動する確率を表し，第 2 項はランダムジャンプによって各ノードへ移動する確率を表す．

2.2.2　ObjectRank

ObjectRank [4] は，PageRank を拡張した，データベース上のオブジェクトの重要性を評価するためのアルゴリズムである．PageRank とは異なり，複数種類のノードとエッジを扱うため，ノードタイプとエッジタイプを考慮する．それぞれのエッジタイプは，そのエッジをたどって遷移する評価値の重みを与えられることによって区別される．

ObjectRank を計算するには，まず Authority Transfer Schema Graph と呼ばれる，グラフの構造とエッジの重みを定義するグラフを構築する．**図 1** に Authority Transfer Schema Graph の一例を示す．Authority Transfer Schema Graph は，評価の対象とするノードタイプの集合と，それぞれのノードタイプ間に存在するエッジタイプの集合で構成される．それぞれのエッジタイプには正方向と逆方向の 2 種類が定義される．これは，重要性を表す評価値は双方向へ遷移すべきであるという考えをもとにしている．たとえば，学術論文の評価値は著者へと遷移すべきであり，逆もまた同様である．しかし，これには例外も考えられる．たとえば，学術論文の評価値は引用元へ遷移すべきであるが，引用先へ遷移すべきでない．各エッジタイプの重みは，それぞれのオブジェクト間の関係をうまく反映するように自由に設定することができる．各ノードタイプから出るエッジタイプの重みの合計は，1 以下である必要がある．

Authority Transfer Schema Graph で定義したグラフの構造やエッジの重みに従って，実際にリンク構造解析の対象とするグラフである Authority Transfer Data Graph を構築する．**図 2** に Authority Transfer Data Graph の一例を示す．Authority Transfer Data Graph は，ノード集合 V とエッジ集合 E によって構成される．すべてのノード，エッジには，それぞれ 1 つのノードタイプ，エッジタイプが対応する．ノード v_i から v_j へのエッジ $e_{ij} \in E$ に与えられる重み $w(e_{ij})$ は以下のように計算する．

$$w(e_{ij}) = \frac{w(e_S)}{OutDeg(v_i, e_S)} \tag{2}$$

ここで，e_S は e_{ij} のエッジタイプであり，$OutDeg(v_i, e_S)$ はノード v_i におけるエッジタイプ e_S の出次数である．

構築した Authority Transfer Data Graph に対して，PageRank と同様の式 (1) を適用し，ObjectRank を計算する．ただし，遷移行列 A の要素 a_{ij} には，ノード v_j から v_i へ

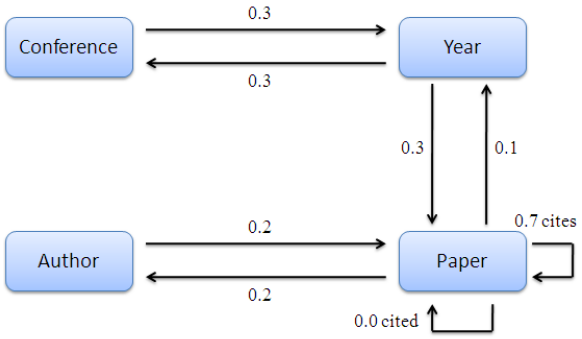

図 1　Authority Transfer Schema Graph
Fig. 1　Authority Transfer Schema Graph.

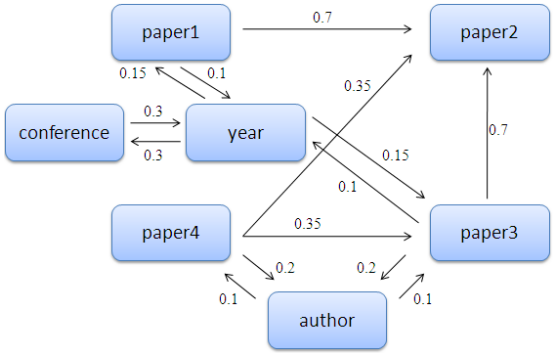

図 2　Authority Transfer Data Graph
Fig. 2　Authority Transfer Data Graph.

のエッジ e_{ji} が存在するときにはその重み $w(e_{ji})$ が格納され，存在しないときには 0 が格納される．

3. 提案手法

ユーザの評価には，フォローによるユーザアカウント自体への評価と，リツイートによるユーザがポストしたツイートへの評価がある．フォローを考慮するだけではユーザが本当に有用な情報を発信しているかを正しく評価することはむずかしい．また，リツイートを考慮するだけではユーザがどのようなツイートをポストする傾向にあるかなど，ユーザアカウント自体の性質を正しく評価することはむずかしい．さらに，従来のようにリツイートの回数を数えるだけでは，ツイートがどのようなユーザにリツイートされ，ユーザの間をどのように伝搬したかをとらえることはできない．そこで，本研究ではフォローとリツイートによる評価をモデル化した User-Tweet Graph にリンク構造解析を適用することでユーザの評価，ランキングを行う手法である TURank を提案する．User-Tweet Graph はユーザとツイートをノードとして持つ．本研究ではリンク構造解析によって抽出した “重要な” ノードが，“有用な” ユーザとなる．リツイートをモデル化し，グラフで表現することで，ツイートがどのようなユーザにリツイートされたか，どのように伝搬したかをとらえることができる．

フォローはユーザからそのユーザがフォローしているユーザへの投票，リツイートはツイートからそのツイートが引用しているツイートへの投票であるとし，以下の 4 つの仮定をおく．

(1)　多くの有用なユーザにフォローされているユーザは有用である．
(2)　多くの有用なツイートにリツイートされているツイートは有用である．
(3)　有用なユーザにポストされているツイートは有用である．
(4)　多くの有用なツイートをポストしているユーザは有用である．

ポストはユーザが得た評価をツイートへ，ツイートが得た評価をユーザへ与えている．これら 4 つの再帰的な仮定は，代表的なリンク構造解析手法である PageRank の概念に類似している．ただし，User-Tweet Graph は，ノードとしてユーザとツイートを，エッジとしてフォロー，ポスト，リツイートを表すため，複数種類のノードやエッジを扱うことが必要となる．このため，PageRank を拡張した ObjectRank を用いる．

本手法は Twitter への適用を目的に提案されたものであるが，次の条件を満たせば他のソーシャルメディアにも適用可能であると考えられる．1 つはユーザ間に何らかの関係があり，ソーシャルグラフを構築できることである．もう 1 つはユーザが生成したコンテンツ間に引用関係があり，引用関係によるグラフを構築できることである．ただし，どちらも正の評価を与えるエッジでなくてはならない．ユーザ間の関係を Twitter におけるフォロー，コンテンツ間の引用関係を Twitter におけるリツイートと読み替えることで，他のソーシャルメディアへの提案手法の適用が可能であると考えられる．

TURank によるユーザ評価の手順を以下に示す．

(1)　グラフの構造やエッジの重みを表すグラフである User-Tweet Schema Graph を定義する．
(2)　User-Tweet Schema Graph で定義したグラフの構造やエッジの重みに従って User-Tweet Graph を構築する．
(3)　構築した User-Tweet Graph から，遷移確率行列を作成する．
(4)　リンク構造解析を行い，ユーザの評価値を算出する．

3.1 節では User-Tweet Schema Graph を定義し，3.2 節で User-Tweet Graph の構築について説明する．3.3 節で遷移行列の作成について説明し，最後に 3.4 節で評価値の算出方法を示す．

3.1 User-Tweet Schema Graph

User-Tweet Graph にその構造や各エッジの重みを与えるために，User-Tweet Schema Graph を定義する[⋆1]．**図 3** に User-Tweet Schema Graph を示す．User-Tweet Schema Graph UTG^S を次のように定義する．

$$UTG^S = (V^S, E^S, \alpha) \tag{3}$$
$$V^S = \{v^S_{user}, v^S_{tweet}\} \tag{4}$$
$$E^S = \{e^S_{follow}, e^S_{followed}, e^S_{post}, e^S_{posted}, e^S_{RT}, e^S_{RTed}\} \tag{5}$$
$$\alpha : E^S \to [0, 1] \tag{6}$$

V^S はノードタイプ集合であり，user タイプノード v^S_{user} と tweet タイプノード v^S_{tweet} からなる．E^S はエッジタイプ集合であり，follow タイプエッジ e^S_{follow}，followed タイプエッジ $e^S_{followed}$，post タイプエッジ e^S_{post}，posted タイプエッジ e^S_{posted}，RT タイプエッジ e^S_{RT}，RTed タイプエッジ e^S_{RTed} からなる．follow タイプエッジはユーザ u から u がフォローするユーザへの関係，post タイプエッジはユーザ u から u がポストしたツイートへの関係，RT タイプエッジはツイート t から t がリツイートするツイートへの関係を表す．また，followed タイプエッジ，posted タイプエッジ，RTed タイプエッジはそれぞれ follow，post，RT エッ

⋆1 User-Tweet Schema Graph は ObjectRank における Authority Transfer Schema Graph に対応する．

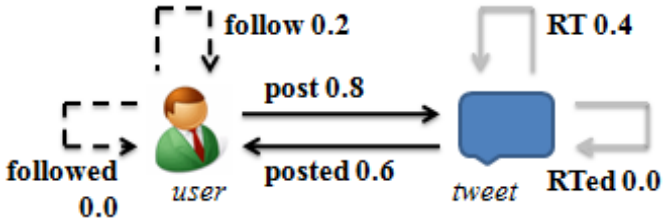

図 3　User-Tweet Schema Graph
Fig. 3　User-Tweet Schema Graph.

ジの逆方向の関係を表す．すなわち，E^S は直積集合 $V^S \times V^S \times \Sigma^*$ の部分集合である．Σ^* はそれぞれのエッジタイプに与えられるラベルの集合である．たとえば，エッジタイプ e^S_{follow} は $(v^S_{user}, v^S_{user}, follow)$ と表される．さらに，α はエッジタイプ集合 E^S から区間 $[0,1]$ への写像であり，各エッジタイプに 0 から 1 の実数値の重みを与える．

各エッジタイプに与えられた重みは，そのエッジタイプによって遷移する評価値の割合を表す．たとえば，図 3 の各エッジタイプの横に示されている重みは一例であるが，各 user タイプノードの評価値の 2 割が follow タイプエッジによって遷移し，8 割が post タイプエッジによって遷移することを示している．followed タイプエッジには重み 0 が与えられているため，評価値が followed タイプエッジによって遷移することはない．PageRank に用いられているランダムサーファモデルを用いて説明すると，あるステップにおいて user タイプノードに存在するランダムサーファは，次のステップには確率 0.2 で follow タイプエッジをたどって次の user タイプノードへ移動し，確率 0.8 で post タイプエッジをたどって次の tweet タイプノードへ移動する．

重みは各エッジの性質を反映するように，手作業によって設定される．ただし，ノードタイプ $v^S \in V^S$ から出るエッジタイプに与えられる重みの合計は 1 以下でなくてはならない．すなわち，次の式を満たさなくてはならない．

$$\sum_{e^S \in OutEdges(v^S)} \alpha(e^S) \leq 1 \tag{7}$$

ここで，$OutEdges(v^S)$ は v^S から出るエッジの集合である．重みの合計が 1 に満たない場合は，ランダムサーファは自己遷移をする．これについては 3.3 節で説明する．

3.2　User-Tweet Graph の構築

User-Tweet Schema Graph で定義したグラフの構造やエッジの重みに従って，リンク構造解析の対象とする User-Tweet Graph を実データから構築する[⋆1]．**図 4** に User-Tweet Graph の一例を示す．User-Tweet Graph UTG は以下のように与えられる．

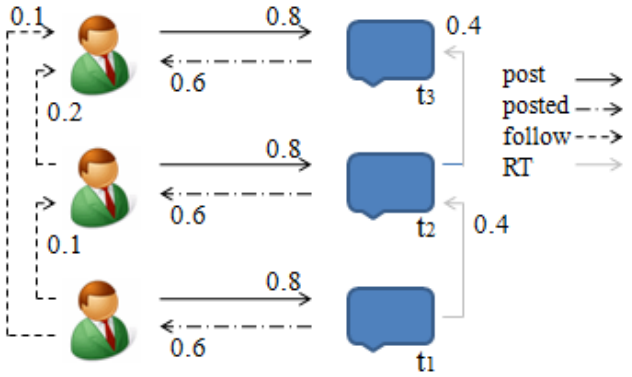

図 4　User-Tweet Graph
Fig. 4　User-Tweet Graph.

$$UTG = (V, E, \lambda, \mu, \beta) \tag{8}$$

$$E \subset V \times V \tag{9}$$

$$\lambda : V \to V^S \tag{10}$$

$$\mu : E \to E^S \tag{11}$$

$$\beta : E \to [0,1] \tag{12}$$

V は User-Tweet Graph が持つノード集合であり，各要素 $v \in V$ のノードタイプは写像 λ によって表される．すなわち，ノード v のノードタイプは V^S に含まれる user タイプノード，tweet タイプノードのいずれかであり，$\lambda(v)$ で表される．また，E は User-Tweet Graph が持つエッジ集合であり，各要素 $e \in E$ のエッジタイプは写像 μ によって表される．すなわち，エッジ e のエッジタイプは E^S に含まれる follow/followed タイプエッジ，post/posted タイプエッジ，RT/RTed タイプエッジのいずれかであり，$\mu(e)$ で表される．さらに，β はエッジ集合 E から区間 $[0,1]$ への写像であり，それぞれのエッジに次のように重みを与える．

$$\beta(e_{ij}) = \frac{\alpha(\mu(e_{ij}))}{OutDeg(v_i, \mu(e_{ij}))} \tag{13}$$

⋆1 User-Tweet Graph は ObjectRank における Authority Transfer Data Graph に対応する．

ただし，$e_{ij} \in E$ はノード v_i からノード v_j へのエッジであり，$OutDeg(v_i, \mu(e_{ij}))$ はノード v_i におけるエッジタイプ $\mu(e_{ij})$ の出次数である．図 4 では，重み 0 のエッジは省略されている．

ここで，表記の簡単化のためにラベル集合 $\Omega = \{user, tweet\}$ と $\Psi = \{follow, followed, post, posted, RT, RTed\}$ を導入する．ノード集合 V とエッジ集合 E は λ と μ を用いて以下のように表記できる．

$$V = \bigcup_{\omega \in \Omega} V_\omega \tag{14}$$

$$E = \bigcup_{\psi \in \Psi} E_\psi \tag{15}$$

$$\forall \omega \in \Omega,\ V_\omega = \{v \in V | \lambda(v) = v_\omega^S\} \tag{16}$$

$$\forall \psi \in \Psi,\ E_\psi = \{e \in E | \mu(e) = e_\psi^S\} \tag{17}$$

V は各タイプのノード集合 V_ω の和集合であり，E は各タイプのエッジ集合 E_ψ の和集合である．V_ω，E_ψ はそれぞれ式 (16)，式 (17) によって与えられる．

User-Tweet Graph は，Twitter におけるユーザとツイートの関係をモデル化している．グラフが user ノードと follow エッジのみで構成されているとすると，そのグラフの構造は PageRank が対象とするグラフと同様である[⋆1]．また，tweet ノードと RT エッジのみで構成されているとしても同様である．しかし，User-Tweet Graph は user ノードと tweet ノードとの間にエッジを有していることにより，user ノードから tweet ノードへ，また tweet ノードから user ノードへ評価値が遷移する．すなわち，User-Tweet Graph では RT エッジによって tweet ノードに集まった評価値が，posted エッジによってそのツイートをポストしたユーザへ遷移する．また同様に，follow エッジによって user ノードに集まった評価値が，post エッジによってそのユーザがポストしたツイートへ遷移する．よって，User-Tweet Graph は，ユーザへの評価とツイートへの評価が相互に影響を及ぼし合うリンク構造解析を可能としている．

また，User-Tweet Graph はリツイートの連鎖を表現している．たとえば，図 4 に示されているように，ツイート t_1 が t_2 をリツイートし，さらに t_2 が t_3 をリツイートしているとき，t_1 の評価値は t_2 を経由して t_3 に遷移する．また，t_2 の評価値が t_3 に遷移するのは明らかである．このように，User-Tweet Graph はツイートがどのようなユーザにリツイートされ，ユーザ間をどのように伝搬したかを表すことができる．

3.3　遷移確率行列の作成

構築した User-Tweet Graph から，各エッジタイプ $\psi \in \Psi$ に対してそれぞれ遷移行列 A^ψ を作成する．A^{follow}，$A^{followed}$ は $|V_{user}| \times |V_{user}|$ 行列，A^{post} は $|V_{user}| \times |V_{tweet}|$ 行列，A^{posted} は $|V_{tweet}| \times |V_{user}|$ 行列，A^{RT}，A^{RTed} は $|V_{tweet}| \times |V_{tweet}|$ 行列である．A^ψ の要素 A_{ij}^ψ は以下の式で与えられる．

$$A_{ij}^\psi = \begin{cases} \beta(e_{ji}) & e_{ji} \in E_\psi \\ 0 & e_{ji} \notin E_\psi \end{cases} \tag{18}$$

すなわち，エッジ $e_{ji} \in E_\psi$ が存在するとき，要素 A_{ij}^ψ にはそのエッジの重み $\beta(e_{ji})$ が格納され，存在しないときは 0 が格納される．

A^ψ の列 k のすべての要素が 0 である場合，すなわち $OutDeg(v_k, e_\psi^S) = 0$ である場合，列 k の各要素に $\alpha(e_\psi^S)/N$ を格納する．N は A^ψ の行の次元である．これはノードから出るあるエッジタイプのエッジが 1 つもない場合，そのエッジタイプの重み分の評価値が失われてしまうのを防ぐための操作である．この操作により，遷移行列 A^ψ の各列の要素の合計は $\alpha(e_\psi^S)$ であることが保証される．

各エッジタイプに対応する遷移行列 A^ψ より，遷移行列 A を作成する．

$$A = \begin{bmatrix} A^{follow} & O \\ A^{post} & A^{RT} \end{bmatrix} + \begin{bmatrix} A^{followed} & A^{posted} \\ O & A^{RTed} \end{bmatrix} \tag{19}$$

ただし，O はすべての要素が 0 である行列である．A^ψ がある特定のエッジタイプのエッジによる遷移のみを表すのに対して，A はすべてのエッジによる遷移を表す $|V|$ 次正方行列である．ノードタイプ v_{user}^S，v_{tweet}^S から発するエッジタイプの重みの合計が 1 でない場合，A の各列の要素の合計は 1 とならない．すなわち，A は遷移確率行列とならない．たとえば，ノードタイプ v_{user}^S から出る各エッジタイプの重みが $\alpha(e_{follow}^S) = 0.4$，$\alpha(e_{followed}^S) = 0$，$\alpha(e_{post}^S) = 0.5$ の場合，A の第 1 列から第 $|V_{user}|$ 列までの各列の要素の合計は 0.9 となるため，A は遷移確率行列ではない．

ここで，自己遷移行列 L を導入する．

$$L = (E - D) \tag{20}$$

$$D_{ii} = \sum_i A_{ij} \tag{21}$$

⋆1 Web グラフは一種類のノードが一種類のエッジによって任意に接続されるグラフ構造である．

E は単位行列であり，D は要素が式 (21) で与えられる対角行列である．L も対角行列となり，各要素には 1 から A の各列の要素の合計を引いたものが格納される．すなわち，L は各ノードの評価値が自己遷移によってどれだけ自らに遷移するかを表す．L を $|V_{user}|$ 次対角行列である小行列 $L_{|V_{user}|}$ と $|V_{tweet}|$ 次対角行列である小行列 $L_{|V_{tweet}|}$ に区分けし（式 (22)）[⋆1]，A と足し合わせることで遷移確率行列 A_N を得る（式 (23)）．

$$L = \begin{bmatrix} L_{|V_{user}|} & O \\ O & L_{|V_{tweet}|} \end{bmatrix} \tag{22}$$

$$\begin{aligned} A_N &= A + L \\ &= \begin{bmatrix} A^{follow} + A^{followed} + L_{|V_{user}|} & A^{posted} \\ A^{posted} & A^{RT} + A^{RTed} + L_{|V_{tweet}|} \end{bmatrix} \end{aligned} \tag{23}$$

3.4　評価値の算出

PageRank を計算する式 (1) から TURank を計算する式を導出し，ユーザの評価値を算出する．式 (1) の評価値ベクトル r を $|V_{user}|$ 次のユーザの評価値ベクトル r^u，$|V_{tweet}|$ 次のツイートの評価値ベクトル r^w を用いて $[r^u, r^w]^T$ と表し，遷移行列 A を 3.3 節で導出した A_N に置き換えることで次の式を得る．

$$\begin{bmatrix} r^u \\ r^w \end{bmatrix} = d \begin{bmatrix} A_L^f & A^{posted} \\ A^{post} & A_L^R \end{bmatrix} \begin{bmatrix} r^u \\ r^w \end{bmatrix} + \frac{(1-d)}{|V|} \begin{bmatrix} u^u \\ u^w \end{bmatrix} \tag{24}$$

ただし，

$$\begin{aligned} A_L^f &= A^{follow} + A^{followed} + L_{|V_{user}|} \\ A_L^R &= A^{RT} + A^{RTed} + L_{|V_{tweet}|} \end{aligned}$$

であり，u^u は要素がすべて 1 の $|V_{user}|$ 次ベクトル，u^w は要素がすべて 1 の $|V_{tweet}|$ 次ベクトルである．式 (24) を各要素ベクトルについて展開し，TURank 方程式 (25) を導出する．

$$\begin{cases} r^u = d(A_L^f r^u + A^{posted} r^w) + \dfrac{(1-d)}{|V|} u^u \\ r^w = d(A^{post} r^u + A_L^R r^w) + \dfrac{(1-d)}{|V|} u^w \end{cases} \tag{25}$$

⋆1 L は対角行列であるため，このように表現できる．

TURank

$r_0^u \leftarrow [1/|V_{user}|, \cdots, 1/|V_{user}|]$
$r_0^w \leftarrow [1/|V_{tweet}|, \cdots, 1/|V_{tweet}|]$
$p \leftarrow 0$
Repeat
　$p \leftarrow p + 1$
　foreach $r_{p,i}^u \in r_p^u$
　　$r_{p,i}^u \leftarrow \sum_{e_{ji} \in E_{follow} \cup E_{followed}} d\beta(e_{ji}) r_{p-1,j}^u + \sum_{e_{ji} \in E_{posted}} d\beta(e_{ji}) r_{p-1,j}^w$
　　$+ dL_{ii} r_{p-1,i}^u + (1-d)/|V|$
　end
　foreach $r_{p,i}^w \in r_p^w$
　　$r_{p,i}^w \leftarrow \sum_{e_{ji} \in E_{RT} \cup E_{RTed}} d\beta(e_{ji}) r_{p-1,j}^w + \sum_{e_{ji} \in E_{post}} d\beta(e_{ji}) r_{p-1,j}^u$
　　$+ dL_{ii} r_{p-1,i}^w + (1-d)/|V|$
　end
until $||r_p^u - r_{p-1}^u||_1 < \epsilon \wedge ||r_p^w - r_{p-1}^w||_1 < \epsilon$
return r_p^u
end

図 5　TURank アルゴリズム
Fig. 5　TURank algorithm.

ユーザの評価値は TURank 方程式を満たす r^u によって与えられる[⋆2]．

TURank 方程式を満たす r^u は**図 5** に示す反復計算アルゴリズムによって計算する．ステップ p における user ノード i の評価値 $r_{p,i}^u$ は，（第 1 項）i へ follow もしくは followed エッジを持つすべての user ノード j のステップ $p-1$ における評価値とエッジの重みの積 $d\beta(e_{ji}) r_{p-1,j}^u$，（第 2 項）$i$ へ posted エッジを持つすべての tweet ノード j のステップ $p-1$ における評価値とエッジの重みの積 $d\beta(e_{ji}) r_{p-1,j}^w$，（第 3 項）自己遷移によって遷移するステップ $p-1$ における自らの評価値 $dL_{ii} r_{p-1,i}^u$，（第 4 項）ランダムジャンプによって得る評価値 $(1-d)/|V|$ の合計である．ただし，第 1 項から第 3 項にはランダムサーファがリンクをたどって遷移する確率 d が掛けられている．tweet ノードの評価値についても user ノードと同様である．この計算をすべてのノードの評価値が収束するまで繰り返す．収束判

⋆2 このとき，TURank 方程式を満たす r^w によって各ツイートにも評価値が与えられる．

定に用いる閾値 ϵ には十分小さい値を設定する．このアルゴリズムを用いて TURank 方程式を満たす r^u を求めることで，ユーザの評価値を算出する．

4. 評価実験

本章では，提案手法の有効性を示すために実施した評価実験について述べる．本実験では，まず提案手法の重みの設定について検証し，次に提案手法と他の主要なユーザ評価手法とを比較する．4.1 節では評価実験に用いたデータの構造や収集方法を説明する．4.2 節では本実験で比較する提案手法の重みについて説明する．4.3 節では実験方法について述べ，4.4 節で結果について考察する．

4.1 実験データ

2010 年 1 月 26 日から 28 日の 3 日間にわたって Twitter API[2] を用いてデータを収集し，本評価実験に用いるデータセット $D = (T, U, F, P, R)$ を構築した．T はツイート集合であり，他のツイートをリツイートしている，または他のツイートにリツイートされている日本語で記述されたツイートを含む．U はユーザ集合であり，T に含まれるツイートを 1 度以上ポストしたユーザを含む．F は follow エッジ集合であり，U に含まれるユーザ同士のすべての follow エッジを含む．P は post エッジ集合であり，ユーザ $u \in U$ から u がポストしたツイート $t \in T$ へのエッジを含む．R は RT エッジ集合であり，ツイート $t_1 \in T$ から t_1 がリツイートしているツイート $t_2 \in T$ へのエッジを含む．データの収集は次の手順で行った．

(1) Twitter Search API[3] を用いて「RT」という文字列を含むツイートを収集し，ツイートを T へ，それらをポストしたユーザを U へ，post エッジを P へ加える[⋆1]．
(2) $t \in T$ がリツイートしているツイートを収集し（詳細は後述する），ツイートを T へ，それらをポストしたユーザを U へ，post エッジを P へ，RT エッジを R へ加える．
(3) Twitter API のメソッド followers/ids を用いて，$u \in U$ のフォロワを収集し，U に含まれるユーザからの follow エッジのみを F へ加える．

上記手順で収集したデータは公式リツイート，非公式リツイートをともに含む．しかし，Twitter API は非公式リツイートに関するデータを提供していないため，上記手順 (2) の RT エッジの収集は独自の方法で行った．まず，ツイート $t \in T$ のテキストに含まれるリツイートされたユーザ名を抽出し，そのユーザの直近 100 ツイートを Twitter API のメソッド statuses/user_timeline を用いて取得する．t のテキストと取得したツイートのテキストとのレーベンシュタイン距離（編集距離）[13] を測り，最も距離の小さい，かつ設定した閾値より小さいツイートを，t にリツイートされたツイートとする．レーベンシュタイン距離が閾値より小さいツイートがなければ，t をデータセットから削除する．

⋆1 Twitter Search API を用いて収集したデータには，ツイートの情報に加えてそれをポストしたユーザの情報も含まれる．

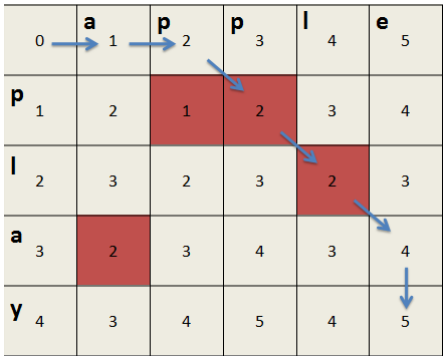

図 6　レーベンシュタイン距離を求めるコスト表
Fig. 6　Cost matrix of Levenshtein distance.

ここで，レーベンシュタイン距離（編集距離）とは，2 つの文字列がどれだけ異なっているかを表す指標である．レーベンシュタイン距離は，片方の文字列を操作してもう片方の文字列に変形するまでの最小の操作コストで定義される．文字列への操作とは，文字の挿入，削除，置換のいずれかを意味し，それぞれの操作には異なるコストを与えることができる．レーベンシュタイン距離は動的計画法を用いて求めることができる．**図 6** は apple と play の間のレーベンシュタイン距離を動的計画法を用いて求めたときのコスト表である．ただし，挿入に 1，削除に 1，置換に 2 のコストが与えられている．左上のマスを始点とし，コスト表を埋めていく．右のマスへの移動は，移動先のマスの列の文字を削除する操作を表す．下のマスへの移動は，移動先のマスの行の文字を挿入する操作を表す．右下のマスへの移動は，移動先のマスの列の文字を行の文字に置換する操作を表す．ただし，色が付けられているマスは行の文字と列の文字が一致するマスであり，置換の際にコストはかからない．各マスにはそのマスへ移動する最小のコストが格納される．これらの操作を左上のマスから順に行っていき，最終的に右下のマスのコストがレーベンシュタイン距離となる．図 6 に示された矢印の経路は，最小の操作コストを与える経路のうちの 1 つであるが，この経路が

表 1　データセットの詳細
Table 1　Dataset details.

	size
# of tweet nodes $\|T\|$	605,968
# of user nodes $\|U\|$	112,035
# of post edges $\|P\|$	605,968
# of RT edges $\|R\|$	369,383
# of follow edges $\|F\|$	14,631,014

示す操作を次に示す．
(1)　pple（a を削除）
(2)　ple（p を削除）
(3)　pla（e を a に置換）
(4)　play（y を挿入）

以上より，apple と play の間のレーベンシュタイン距離は 5 と求められる．

RT エッジの収集では，挿入コストには削除コストに比べて大きな値を設定した[⋆1]．これは，Twitter の文字数制限により，元のツイートを短くするためにテキストの一部を削除してリツイートする傾向があるためである．

以上の手順によって収集したデータから構築したデータセット D の詳細を**表 1** に示す．本実験においては限定的なデータを用いているが，提案手法は大規模なデータに対しても適用可能である．提案手法は PageRank と同じ行列計算に帰着することができるため，Google[⋆2]が大規模なデータに対する PageRank の計算に用いている MapReduce フレームワークを用いることができる．ツイートは爆発的な速さで増えていくが，ツイート間のエッジを表すリツイートは時間的に非常に近いツイート間でのみ発生する傾向にあるため[11]，ノードに対してエッジは非常に少ない．そのため，計算量は Web グラフと比べて少ないと考えられる．

4.2　重みの設定

提案手法を Twitter，あるいは他のソーシャルメディアに適用する際には，User-Tweet Schema Graph の重みの設定が重要である．そのため，本実験では提案手法の適当な重みの設定について検証するために，**表 2** に示す重みを設定した 4 つの User-Tweet Schema Graph を用いてそれぞれランキングを作成し比較する．比較対象とした 4 つの重み設定の

⋆1 置換コストには削除コストと挿入コストの和を設定した．
⋆2 http://google.com

表 2　TURank weights
Table 2　TURank weights.

	follow	followed	post	posted	RT	RTed
TURank1	0.4	0.0	0.6	0.6	0.4	0.0
TURank2	0.2	0.0	0.8	0.6	0.4	0.0
TURank3	0.2	0.0	0.8	0.4	0.6	0.0
TURank4	0.2	0.0	0.8	0.6	0.2	0.2

根拠は次のとおりである．TURank1 から TURank3 の重みは，RT エッジと follow エッジの重みの割合がどのようなときに提案手法が有効かを検証するためのものである．また，TURank4 は RTed エッジへ 0 より大きな重みを設定することの有効性を検証するためのものである．すべての場合において followed エッジには重み 0 を設定したが，これは学術論文の評価値が引用元から引用先へ遷移すべきではないということと同様に，ユーザの評価値はフォロワへ遷移すべきではないという考えに基づく．また提案手法では，リツイートとフォローがユーザ評価の指標となっているため，RT/RTed エッジ，follow/followed エッジの重み設定が重要であり，post/posted エッジの重み設定は結果にあまり強い影響は及ぼさないと考えられる．そのため，まず RT/RTed エッジ，follow/followed エッジの重みを決定した後に，各ノードタイプから発するエッジタイプの重みの合計が 1 になるように post/posted エッジの重みを副次的に決定した．

4.3　実験方法

本実験では，4.2 節で決定した重みによる提案手法の比較と，提案手法と他の主要なユーザ評価手法との比較の 2 つの実験を行う．比較対象とする手法は次の 5 手法である．

- PageRank
- HITS[9]
- FollowNum
- RTNum
- FollowRT

PageRank と HITS は，それぞれのアルゴリズムをソーシャルグラフに適用し，そのスコアによってユーザをランキングする．FollowNum はユーザが持つフォロワ数によってユーザをランキングする．RTNum はユーザがポストしたツイート群が得たリツイート数の合計によってユーザをランキングする．FollowRT はフォロワ数とリツイート数の線形和によってユーザをランキングする．線形和におけるそれぞれの重みは 0.5 とした（理由は後述する）．

重みを変えた 4 つの提案手法と，比較対象とする 5 手法によって 9 つのランキングを作成し，34 名の被験者による評価を行った．それぞれのランキングの上位 25 ユーザを評価対象とし，被験者は評価対象ユーザの有用性について 1 から 5 の 5 段階で評価した．5 段階の評価基準は次のとおりである．

- 1：まったく有用ではない
- 2：有用ではない
- 3：どちらかといえば有用である
- 4：有用である
- 5：非常に有用である

被験者によって評価対象ユーザに与えられた 1 から 5 のスコアを評価値と呼ぶ．ユーザの評価はそのユーザの最新の 100 ツイートを閲覧することで行う．ユーザを評価するにあたって，有用なユーザを以下のように定義した．

- 多くのユーザが注目する最新のニュースや話題を発信するユーザ．
- 何らかの話題について，多くのユーザに影響を与えるような意見を発信するユーザ．
- 多くのユーザに面白い（ユーモアがある，ためになるなど）と見なされるツイートを発信するユーザ．

本実験では 34 名の被験者によって与えられた評価値の平均が 3 以上となるユーザの集合を正解セットとする．正解セットに含まれるユーザを適合とし，各ランキングの適合率を算出する．**図 7**，**図 8** は上位 k ユーザの適合率の平均をプロットしたグラフである．図 7 は重みを変えた 4 つの提案手法を比較したグラフであり，図 8 は提案手法と他の 5 つの手法とを比較したグラフである．

適合率の比較に加えて，被験者によって与えられた評価値そのものの比較も行う．**表 3** は各手法によるランキングの NDCG の値を示している．NDCG とは，ランキング $(u_1, u_2, \cdots, u_{25})$ に対して，式 (26) で与えられる指標である．

$$NDCG = \frac{DCG}{IDCG} \tag{26}$$

$$DCG = score(u_1) + \sum_{i=2}^{25} \frac{score(u_i)}{\log_2 i} \tag{27}$$

ここで，$score(u_i)$ は被験者によってユーザ u_i に与えられた評価値の平均値であり，IDCG は理想的なランキングに対する DCG である．本実験では IDCG は全ユーザを $score(u)$ の降順でソートしたランキングに対する DCG となる．NDCG はランキング上位に $score(u)$

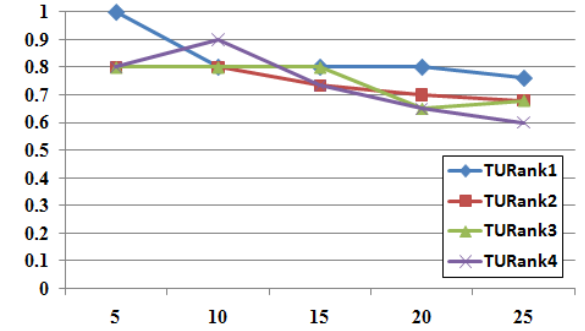

図 7　重みを変えた提案手法の適合率
Fig. 7　Precision of proposed methods using varied weights.

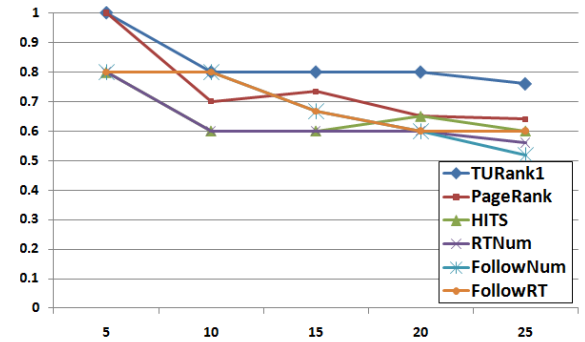

図 8　TURank と他手法の適合率
Fig. 8　Precision of the proposed method and existing methods.

表 3　各手法の NDCG 値
Table 3　NDCG of all methods.

	NDCG
TURank1	0.887
TURank2	0.873
TURank3	0.866
TURank4	0.850
PageRank	0.867
HITS	0.838
FollowNum	0.798
RTNum	0.769
FollowRT	0.820

の大きいユーザ u が位置するほど値が大きくなる．次節ではこれらの結果に対する考察を行う．

4.4 考　　察

本節では結果についての考察を行う．4.4.1 項では重みを変えた 4 つの提案手法を比較した結果について考察し，4.4.2 項では提案手法と他の 5 つの手法とを比較した結果について考察する．

4.4.1 重みを変えた提案手法の比較

図 7，表 3 によると，適合率，NDCG ともに TURank1 が最も高い有効性を示している．表 1 によると，RT エッジは follow エッジに比べて格段に少ない．そのため，TURank2 や TURank3 のように follow エッジに比べて RT エッジに比較的大きな重みを与えてしまうと，グラフの疎な部分が結果に大きな影響を与えることになる．これは，一時的なリツイートの増加など，局所的な影響を強く受けてしまうことを意味する．TURank2 と TURank3 によるランキングでは，実験に用いたデータ期間にのみ一時的に多くのリツイートを得ているが，その後あまりリツイートされていないユーザが上位に位置していた．これらのユーザは被験者によって低い評価値を与えられたため，結果として TURank2，TURank3 は TURank1 と比べて低い有効性を示したと考えられる．

また，RTed エッジに 0 より大きな重みを与えることの有効性を検証するために設定された TURank4 は，適合率，NDCG ともに最も低い結果を示している．RTed エッジに 0 より大きな重みを与えると，リツイートされたツイートからリツイートしたツイートへスコアが伝搬する．そのため，有用なツイートを多くリツイートするようなユーザが上位にランキングされることが期待された．しかし，結果として有用とはいえない 2 つのボット[⋆1]が上位に抽出された．1 つはランダムに夢についてのツイートをリツイートする yumemitter というボットであり，もう 1 つは醤油についてのツイートをランダムにリツイートする soysoucebot というボットであった．積極的にリツイートを行うユーザを上位に抽出したのは妥当な結果であるが，これらのユーザは被験者によって低い評価値を与えられたため，TURank4 は低い有効性を示した．この結果から，Twitter 上には有用な情報を積極的にリツイートするユーザがあまりいないため，上記の 2 つのボットしか抽出できなかったのではないかと考えられる．

以上の考察により，Twitter あるいは他のソーシャルメディアに提案手法を適用する際には，次のような指針によって重みを設定すればよいと考えられる．

- リツイートが行われる頻度に応じて RT エッジと follow エッジの重みを設定する．現在の Twitter ではフォローによるソーシャルグラフに比べてリツイートによる情報伝搬のグラフは非常に疎であったため，RT エッジの重みを比較的小さく設定したときに有効であった．しかし，提案手法を適用するソーシャルメディアにおいては，コンテンツの引用がどれだけ行われているかを考慮して RT エッジと follow エッジの重みの比を設定する必要がある．
- 現在の Twitter 上には有用な情報を積極的にリツイートし，伝搬させることを目的とするユーザはあまりいないため，RTed エッジに 0 より大きな重みを与えると有用ではないユーザが上位に抽出され，良い結果にはならなかった．提案手法を適用するソーシャルメディアに有用な情報を厳選して多く伝搬させるユーザが存在する場合には，RTed エッジに 0 より大きな重みを与えることでそのようなユーザの抽出が可能であると考えられる．

この結果をうけて，最も有効であった TURank1 と他の手法とを比較した．また，最も有効であった TURank1 の follow エッジの重みと RT エッジの重みは 1 : 1 であったため，FollowRT の線形和におけるフォロワ数とリツイート数のそれぞれの重みには 0.5 を与えた．

4.4.2 他の手法との比較

図 8，表 3 によると，TURank によるランキングが最も高い有効性を示している．提案手法に続いて，グラフ解析を用いた PageRank，HITS がある程度高い有効性を示し，単純にフォローとリツイートの回数を数える 3 手法が最も低い有効性を示している．**表 4**，**表 5**，**表 6**，**表 7**，**表 8**，**表 9** に各手法によるランキング結果を示す．ランキング結果について以下のように考察する．

フォローとリツイートの回数を数える 3 手法が低い有効性を示した理由を考察する．この結果は単純にフォロー，リツイートの回数を数えてユーザを評価するだけでは不十分であることを示していると考えられる．これらの手法では，どのユーザにフォローされたか，またどのツイートにリツイートされたかなど，フォロー元，リツイート元のノードのスコアを考慮できていない．そのため，yahoo_shopping や shuumai などの不特定多数のフォローまたはリツイートを集めるボットを上位に抽出してしまったのではないかと考えられる．これらのボットは被験者によって低い評価値を与えられている．また，RTNum によって上位に抽出された nelson_koenji は大喜利のお題をポストし，リツイートを募るユーザであるが被験者によって低い評価をされている．これらのユーザはグラフ解析を用いる手法では上位

⋆1 ボットとはプログラムによって自動的にポストを行うユーザのことである．

表 4　TURank1 によるランキング結果
Table 4　Ranking by TURank1.

順位	ユーザ名
1	masason
2	555hamako
3	takapon_jp
4	kazuyo_k
5	astro_soichi
6	hikaruijuin
7	mainichijpedit
8	shuzo_matsuoka
9	asahi
10	kohmi
11	meigenbot
12	shakase
13	tsuda
14	kharaguchi
15	jaxa_jp
16	hmikitani
17	nhk_pr
18	shiro_tsubuyaki
19	47news
20	renho_sha
21	seikoito
22	hazuma
23	skmt09
24	taguchi
25	samfurukawa

表 5　PageRank によるランキング結果
Table 5　Ranking by PageRank.

順位	ユーザ名
1	masason
2	555hamako
3	takapon_jp
4	kazuyo_k
5	jaxa_jp
6	comic_natalie
7	astro_soichi
8	owarai_natalie
9	hikaruijuin
10	mainichijpedit
11	hmikitani
12	kohmi
13	47news
14	kharaguchi
15	asahi
16	toriaezu
17	tsuda_pr
18	47newsflush
19	seikoito
20	shakase
21	skmt09
22	shiro_tsubuyaki
23	renho_sha
24	shuzo_matsuoka
25	room66plus

表 6　HITS によるランキング結果
Table 6　Ranking by HITS.

順位	ユーザ名
1	masason
2	takapon_jp
3	kazuyo_k
4	555hamako
5	tsuda
6	kohmi
7	inadatomoyuki
8	note_man
9	bonbokorin
10	mainichijpedit
11	q2e2d2
12	hmikitani
13	hikaruijuin
14	astro_soichi
15	asahi
16	renho_sha
17	sentan
18	ohanika
19	kharaguchi
20	taguchi
21	sasakitoshinao
22	shuzo_matsuoka
23	omowaku
24	makeplex
25	nobi

表 7　FollowNum によるランキング結果
Table 7　Ranking by FollowNum.

順位	ユーザ名
1	mooris
2	gachapinblog
3	tenkijp
4	takapon_jp
5	asahi
6	mainichijpedit
7	twj
8	kazuyo_k
9	yahoo_shopping
10	taguchi
11	kotoripiyopiyo
12	kohmi
13	suadd
14	kengo
15	kogure
16	nobi
17	abfly
18	msugaya
19	fshin2000
20	tokuriki
21	matsuyou
22	taromatsumura
23	ryotheskywalker
24	natalie_mu
25	rkmt

表 8　RTNum によるランキング結果
Table 8　Ranking by RTNum.

順位	ユーザ名
1	hazuma
2	meigenbot
3	shuzo_matsuoka
4	meinichijpedit
5	iwakamiyasumi
6	47news
7	itmedia_news
8	shuumai
9	nelson_koenji
10	hikaruijuin
11	nhk_pr
12	hokayan
13	kotoba_bot
14	kazuyo_k
15	idanbo
16	samfutukawa
17	masason
18	shakase
19	eguchinn
20	kenjieno
21	akhk
22	astro_soichi
23	knnkanda
24	gotch_akg
25	katokichicoltd

表 9　FollowRT によるランキング結果
Table 9　Ranking by FollowRT.

順位	ユーザ名
1	mainichijpedit
2	hazuma
3	meigenbot
4	kazuyo_k
5	shuzo_matsuoka
6	asahi
7	mooris
8	natalie_mu
9	takapon_jp
10	iwakamiyasumi
11	hikaruijuin
12	47news
13	itmedia_news
14	kogure
15	nobi
16	shuumai
17	gachapinblog
18	nelson_koenji
19	taguchi
20	abfly
21	nhk_pr
22	hokayan
23	kotoba_bot
24	kohmi
25	masason

に抽出されることはなかった．さらに，matsuyou や suadd は多くのフォロワを集める著名人であるが，低い評価値を与えられている．この 2 ユーザがグラフ解析を用いた手法で上位に抽出されていないのも特徴的である．

RTNum は hazuma や kenjieno のような，会話を目的としてリツイートを用いるユーザを上位へ抽出してしまっている．これらのユーザは低い評価値を与えられている．会話を目的としたリツイートは明らかにツイートへの投票であるとはいえないが，RTNum ではそれとフォロワへのツイートの伝搬を目的としたリツイートとを区別することができていない．会話を目的としたリツイートは会話に参加する数名のユーザ間で行われるが，ツイートの伝搬を目的としたリツイートは多くのユーザ間を伝搬していく．ユーザ間を広く伝搬するツイートは木構造のように様々なツイートからリツイートされるため，グラフ解析を用いた手法では大きなスコアを得ることになる．しかし，会話を目的としたリツイートは特定のユーザ間でのみ行われるため，あまり大きなスコアを得ることはない．そのため，提案手法では会話を目的としたリツイートとツイートの伝搬を目的としたリツイートとを区別することができていると考えられる．

ryotheskywalker や eguchinn などの，フォローまたはリツイートのどちらか一方のみが多いユーザは低い評価を与えられる傾向にあった．そのため，フォローとリツイートの線形

和を用いた FollowRT は，どちらかのみを用いた FollowNum や RTNum より高い有効性を示していると考えられる．FollowRT は提案手法よりは低い値を示しているものの，ユーザ評価の指標としてリツイートを用いることの有効性を示していると考えられる．

グラフ解析を用いた 3 手法のうち，提案手法が最も高い有効性を示した理由を考察する．提案手法はリツイートされた数は多いがあまりフォローされていない nhk_pr や meigenbot などのユーザを抽出している．また，mainichijpedit や asahi などのニュースアカウントや，astro_soichi などの著名人ユーザの順位が少しずつ上昇している．これらのユーザは非常に大きな評価値を与えられている．PageRank や HITS ではユーザがどれだけ多くリツイートされていても，多くのフォローを得ていない限り上位に抽出することはできない．そのため，多くのリツイートを得るユーザの抽出という点で提案手法は有効であるといえる．HITS は PageRank に比べて低い有効性を示しているが，これは HITS におけるハブとオーソリティの概念が Twitter には適さないためだと考えられる．HITS によるランキング結果によると，単に多くのユーザをフォローしているユーザが良いハブとなってしまっていた．これは，良いオーソリティへのエッジを多く持つノードは良いハブであるという前提に反する．

フォローをユーザ評価の指標とした手法では，takapon_jp（堀江貴文氏）や kazuyo_k（勝間和代氏），masason（孫正義氏）などの明らかな著名人が上位に位置する傾向にあった．一方，リツイートをユーザ評価の指標とした手法では，meigenbot や kotoba_bot などの，ユーザ間を広く伝搬するような名言をポストするユーザや，samfurukawa や akhk などの明らかな著名人ではないが，注目を集めるツイートをポストするユーザなどが上位に抽出された．Kwak ら[11] は本稿と同様に，フォローによるユーザランキングとリツイートによるユーザランキングは大きく異なると報告している．このように，どちらの手法によるランキングにも有用なユーザは存在し，それぞれの手法によって上位に抽出できるユーザは異なるため，2 つの指標を取り入れた手法が必要とされると考えられる．

5. 関 連 研 究

近年，Twitter に関する研究が多く行われている．Java ら[8] はソーシャルグラフを解析することにより，ユーザの地理的特性について調査している．また，ユーザが形成するコミュニティについても調査し，ユーザを 3 つのカテゴリに分類している．Huberman ら[7] は，フォローによって構築されるソーシャルグラフはユーザ間の関係をうまく表現していないとし，独自に定義した友人関係によって構築されるソーシャルグラフがうまくユーザ間の関係を表していることを示している．Krishnamurthy ら[10] は，ユーザが持つフォロワ数と，ユーザがフォローしているユーザの数との関連を調べ，ユーザの特性について研究している．Boyd ら[5] は，リツイートの利用形態を，リツイートの記述形式，リツイートの目的，リツイートの対象という観点から調査している．

ツイートの伝搬を取り上げた研究もいくつか行われている．Ye ら[18] は，リプライによるツイートの伝搬を研究し，それらは広く，速く伝わり，伝搬が短い期間で収束すると報告している．リプライとは指定したツイートに向けてツイートをポストすることであり，“返信” のようなものである．また，特定の話題に関するツイートの伝搬に限った調査も行い，その話題についての情報を発信するユーザ数と，それらを受信するユーザ数との関係性について述べている．Kwak ら[11] は，リツイートによるツイートの伝搬について研究し，ユーザが持つフォロワ数と，そのユーザのツイートがリツイートされ，どれだけの数のユーザのもとへ伝搬するかの間には関連がないことを報告している．また，有用なツイートは 1 度リツイートされると，短い期間のうちに連鎖的にリツイートされ，ユーザの間を伝わっていくことを明らかにした．

Twitter ユーザを評価し，ランキングする手法の研究も多く行われている．Weng ら[16] は，ユーザのポスト数やユーザ間の類似度を考慮し，PageRank を用いてソーシャルグラフを解析することでユーザの有用性を測る手法 TwitterRank を提案している．TURank は TwitterRank とは以下の点で異なる．1 つは，ソーシャルグラフではなく，リツイートをモデル化した User-Tweet Graph を解析している点である．もう 1 つは，ツイートの内容は考慮しておらず，完全にグラフベースの手法である点である．TwitterRank はトピックに基づくユーザの評価を行っているため，今回の評価実験では比較対象としていない．Leavitt ら[12] は，フォローのみを考慮したユーザの評価は不十分であるとし，リプライやリツイートなどのユーザ同士のコミュニケーションを考慮した手法を提案している．Cha ら[6] も同様にリプライやリツイートによるユーザの評価を行い，フォローによる評価との比較実験を行っている．これらの手法はリプライやリツイートの数に基づく評価を行っており，リンク構造を考慮していないため，本研究とは異なる．

6. 結　　論

本稿では，有用なユーザの発見を目的とし，Twitter ユーザのランキング手法を提案した．提案手法はフォローによるユーザへの評価とリツイートによるツイートへの評価の両方を考慮し，ランキングを行う．提案手法がグラフ解析の対象とする User-Tweet Graph は，ツイートがどのようなユーザにリツイートされ，どのようにソーシャルグラフ上を伝搬してい

るかを適切に表現している．

評価実験の結果により，提案手法が既存手法に比べて有効であることが示された．多くのフォローを集めるユーザと多くのリツイートを集めるユーザは異なる傾向にあったが，どちらも被験者によって有用であるとされた．しかし，フォローまたはリツイートのみを考慮するランキング手法ではそれら両方のユーザを上位に抽出することはできない．提案手法はユーザ評価の指標としてフォローとリツイートの両方を考慮することで，両方のユーザ群を上位に抽出することができた．

ユーザの有用性はトピックによって異なる．たとえば，テニスについて詳しいユーザはその分野においては有用であるが，料理の分野において有用であるとは限らない．よって，今後の課題としてユーザのトピックを考慮した評価があげられる．User-Tweet Graph はコンテンツを持つ tweet ノードを含むため，そのトピックを同定し，user ノードへ遷移させることでユーザのトピックの考慮が可能であると考えられる．

謝辞　本研究の一部は科学研究費補助金特定領域研究（#21013004）による．

参 考 文 献

1) Twitter. http://twitter.com
2) Twitter API. http://apiwiki.twitter.com/Twitter-API-Documentation
3) Twitter Search API. http://apiwiki.twitter.com/w/page/22554756/Twitter-Search-API-Method:-search
4) Balmin, A., Hristidis, V. and Papakonstantinou, Y.: ObjectRank: Authority-Based Keyword Search in Databases, *VLDB* (2004).
5) Boyd, D., Golder, S. and Lotan, G.: Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter, *Hawaii International Conference on System Sciences*, Vol.0, pp.1–10 (2010).
6) Cha, M., Haddadi, H., Benevenuto, F. and Gummadi, K.P.: Measuring user influence in Twitter: The million follower fallacy, *ICWSM 2010: Proc. International AAAI Conference on Weblogs and Social Media* (2010).
7) Huberman, B.A., Romero, D.M. and Wu, F.: Social networks that matter: Twitter under the microscope, *1st Monday*, Vol.14, No.1 (Jan. 2009).
8) Java, A., Song, X., Finn, T. and Tseng, B.: Why We Twitter: Understanding microblogging usage and communities, *Joint 9th WEBKDD and 1st SNA-KDD Workshop*, San Jose, CA (2007).
9) Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment, *Proc. 9th ACM SIAM Symposium on Discrete Algorithms* (*SODA '98*), pp.668–677 (1998).
10) Krishnamurthy, B., Gill, P. and Arlitt, M.: A few chirps about twitter, *Proc. 1st Workshop on Online Social Networks*, ACM (2008).
11) Kwak, H., Lee, C., Park, H. and Moon, S.: What is Twitter, a social network or a news media?, *World Wide Web Conference* (2010).
12) Leavitt, A., Burchard, E., Fisher, D. and Gilbert, S.: The influentials: New approaches for analyzing influence on twitter, *A publication of the Web Ecology Project* (2009).
13) Levenshtein, I.V.: Binary Codes capable of correcting deletions, insertions, and reversals, *Cybernetics and Control Theory*, Vol.10, No.8, pp.707–710 (1966).
14) Moore, R.J.: *New data on Twitter's users and engagement* (2010). http://themetricsystem.rjmetrics.com/2010/01/26/new-data-on-twitters-users-and-engagement/
15) Page, L., Brin, S., Motwani, R. and Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web, Technical Report 1999-66, Stanford InfoLab. (1999).
16) Weng, J., Lim, E., Jiang, J. and He, Q.: TwitterRank: Finding Topic-sensitive Influential Twitterers, *WSDM* (2010).
17) Yamaguchi, Y., Takahashi, T., Amagasa, T. and Kitagawa, H.: TURank: Twitter User Ranking Based on User-Tweet Graph Analysis, *WISE*, pp.240–253 (2010).
18) Ye, S. and Wu, S.F.: Measuring Message Propagation and Social Influence on Twitter.com, *SocInfo 2010* (2010).

(平成 22 年 12 月 20 日受付)
(平成 23 年 4 月 3 日採録)

(担当編集委員　小山 聡)

山口　祐人

2010 年筑波大学第三学群情報学類卒業．現在，同大学院システム情報工学研究科に在学中．データマイニング，情報検索等に関する研究に従事．日本データベース学会学生会員．

天笠　俊之（正会員）

1994 年群馬大学工学部情報工学科卒業．1999 年同大学院工学研究科修了．博士（工学）．奈良先端科学技術大学院大学情報科学研究科助手，筑波大学大学院システム情報工学研究科講師を経て，2009 年 12 月より同准教授（計算科学研究センター准教授を兼任）．宇宙航空研究開発機構宇宙科学研究所宇宙科学情報解析研究系客員准教授．XML データベース，e サイエンスにおけるデータベース応用等の研究に従事．日本データベース学会，電子情報通信学会，ACM，IEEE Computer Society 各会員．

髙橋　　翼

2008 年筑波大学第三学群情報学類卒業．2010 年同大学院システム情報工学研究科博士前期課程修了．修士（工学）．現在，日本電気株式会社勤務．データマイニング，情報検索等の研究に従事．

北川　博之（フェロー）

1978 年東京大学理学部物理学科卒業．1980 年同大学院理学系研究科修士課程修了．日本電気（株）勤務の後，1988 年筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学大学院システム情報工学研究科教授，ならびに計算科学研究センター教授．理学博士（東京大学）．異種情報源統合，XML とデータベース，データマイニング，センサデータベース，WWW データ管理等の研究に従事．著書『データベースシステム』（昭晃堂），『The Unnormalized Relational Data Model』（共著，Springer-Verlag）等．日本データベース学会理事，電子情報通信学会フェロー，ACM，IEEE-CS，日本ソフトウェア科学会各会員．