

Twitter User Recommender

～Twitterのお気に入り機能を用いたおすすめユーザ推薦システム～

渡部 将太[†] 宮森 恒[†]

[†] 京都産業大学コンピュータ理工学部

〒 603-8555 京都府 京都市北区上賀茂本山

E-mail: †{g0847506,miya}@kyoto-su.cse.ac.jp

あらまし 近年マイクロブログサービスが世界中で利用されている。特に、Twitterでは、フォロー機能により、ユーザは、自分の興味のある他のユーザのつぶやきを随時確認することができる。しかし、多くのユーザをフォローすると画面表示が煩雑になるという問題があるため、自分に適するユーザを効率よく絞り込む手法が重要となる。従来のユーザ推薦システムは、フォロー関係を利用したものが多く、フォロー関係はあるユーザから他のユーザへの評価の2値表現に過ぎず、これのみではユーザの評価度合の大小を直接表現することはできない。本稿では、つぶやき毎に評価の有無を付与できる、お気に入り機能に着目し、これに基づくユーザ毎の評価を考慮した推薦システムを提案する。これにより、フォロー関係では発見が困難だった局所的なユーザ間での評価の高いユーザを発見できることが期待される。予備実験により、想定したユーザを見つけやすいという感触が得られた。

キーワード マイクロブログ, Twitter, ユーザ推薦, Favstar, お気に入り機能

Twitter User recommender

～The user recommendation system using the Favorite function of Twitter～

Shouta WATABE [†] and Hisashi MIYAMORI [†]

[†] Faculty of Computer Science and Engineering, KyotoSangyo University

KamigamoMotoyama, Kita, Kyoto, 603-8555 Japan

E-mail: †{g0847506,miya}@kyoto-su.cse.ac.jp

Abstract Recently, several microblog services have been expanding their shares in the world. In Twitter, users can efficiently browse Tweets posted by other users of interest at any time by using Follow function. However, there is a problem that the display becomes easily complicated when users follow too many users. So, the methods become important to efficiently narrow down users when deciding which users are to be followed. Conventional user recommendation systems often utilize Follow relations, each of which is represented as the existence of appreciation from one user to another as binary expression, meaning that it is hard to directly estimate the degree of appreciation from one user to another. In this paper, a user recommendation system is proposed considering the degree of appreciation from a user to another based on Favorite function where users can give binary appreciation to each Tweet. It is expected to provide easier access to highly appreciated users among local users, which have been difficult to be found when using only Follow relations. Preliminary experiments showed a positive response of easier discovery of intended users.

Key words microblogging, Twitter, User recommender

1. はじめに

マイクロブログサービスとは短い文章を時間軸に沿って表示するブログの一種である。その中の Twitter と呼ばれるサー

ビスが世界各国でシェアを伸ばしており、現在約2億アカウントが存在していると発表されている [1]。また、その際 Twitter で使用されている言語のうち約 25 % が日本語であると調査発表されている [1]。Twitter ユーザは、ツイートと呼ばれる 140

文字までの短い文章を発信し、タイムラインと呼ばれる自身のインタフェース上に時系列順に表示するだけでなく、フォロー機能を用いることにより、他のユーザのツイートを自分のタイムライン上に常時表示させることができる。フォロー機能により、自分が気になるユーザやイベント等の最新の情報を手軽に確認できる利点がある。一般に、フォローするユーザを増やしていく際には、図1に示す手順を踏むことが多い。候補となるユーザを絞り込み、そのユーザのプロファイルやツイート内容からフォローするかどうかを自由に判断する。しかし、2億人以上のアカウントが存在しているTwitterから自分の興味のあるユーザを探し、フォローの可否を見定めるのは容易ではない。

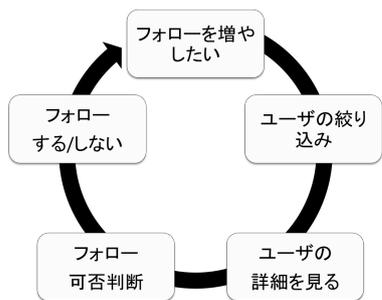


図1 一般的なフォロー作業のイメージ

そこで、Twitter ユーザは、フォローをする際に2億以上のアカウントから自分に見合ったユーザを絞り込む際に、以下のような情報を参照することが多い(図2)。

- Twitter 公式に存在するおすすめユーザ群やカテゴリ分類された有名人
- ワード検索で表示されたり、公式 RT で流れてきたユーザ群
- 自分をフォローしているユーザ群
- ネットサーフィン等を行う際の巡回サイトの管理人やサイトの更新をツイートするアカウント
- 実生活における友人などからの口コミ

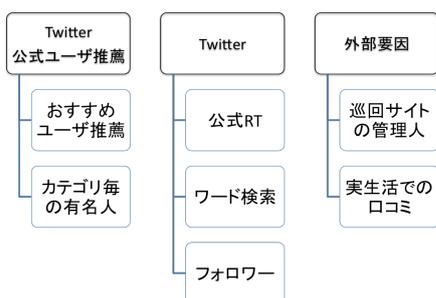


図2 フォロー先ユーザの絞り込みの一例

しかし、Twitter 公式のユーザ推薦はそのユーザが周りのユーザからどれだけ評価されているか、また誰と仲がいいのかといった情報が考慮されていない問題があると考えられる。単純な協調フィルタリングでユーザのフォロー関係を見ていくと、サイトの更新をつぶやき続ける bot や有名人ばかりが上位に上がってしまい、自分のフォローしているユーザ間で人気のあ

るユーザ等を調べる事が出来ないのである。つまり、フォロー関係はあるユーザから他のユーザへの評価の2値表現に過ぎず、これのみではユーザの評価度合の大小を直接表現することは困難であると考えた。本稿では、つぶやき毎に評価の有無を付与できる、お気に入り機能に着目し、これに基づくユーザ毎の評価を考慮した推薦システムを提案する。これにより、フォロー関係では発見が困難だった局所的なユーザ間での評価の高いユーザを発見できることが期待される。

2. 関連研究

Twitter のおすすめユーザ推薦は様々な物が存在している。Twitter の公式サイトでも、おすすめユーザ検索 [2] は実装されている。アルゴリズムなどは何も公開されていないが、ユーザの下にこのユーザをフォローしているユーザは○○、◇◇、△△といったような情報が出てくるため、この推薦はフォロー関係をもとにしていると思われる。また公式サイトしか行えない推薦として、グローバル IP アドレス等をもとにおすすめユーザを推薦する手法なども実装されたことがある。また、おすすめユーザ推薦以外に、自分と似ているユーザを表示する物も Twitter 公式サイトで公開されている [3]。これら以外にも Twitter 公式サイトではメールアドレスや、ハンドルネームでユーザを検索する事が出来る機能が存在する。これら全ておすすめユーザを提示する物だが、本稿と推薦に使用する情報が恐らく違っている点が違う。

Twitter 公式と同じように、フォロー情報を用いたユーザ推薦手法も存在する [4]。この研究ではユーザがフォローをする時に自分の中の属性の一つが伝搬すると考えるフォローによる属性伝搬モデルを使用してフォロー先推薦を行っている。フォロー関係を用いた属性伝搬モデルとは、ユーザがユーザをフォローする時にフォローする側の属性の一つがフォロー先のユーザに伝搬するという考え方である。このユーザ属性の伝搬は本稿のユーザ層選定の考えと似通っている。

本稿がおすすめユーザを選定する際に使用するデータであるお気に入りのデータを利用した研究もまた存在している。[5] この研究はツイートを、広く一般の人にとって情報として有用なものとそうでないものに分類する手法を提案し、ユーザの属性をある程度推定する研究である。本稿と同じくお気に入り機能を利用しているが、ユーザのスコア計算に HITS アルゴリズムを使用し、お気に入り1つ1つに対しても価値の差を認める部分が本稿と異なっている研究である。

3. お気に入り機能

Favorite 機能は日本語版 Twitter ではお気に入り機能と翻訳されている機能で、あるユーザが、別のユーザが発言した1件のツイートに、それを気に入った証として、星印「☆」を1つだけ付与できるという機能である。日本では、このお気に入りの星印を付与する行為は「ふぁぼり」、星印を付与されることは「ふぁぼられ」と呼ばれることが多い。本稿では、あるユーザと直接の「ふぁぼり」あるいは「ふぁぼられ」の関係を持つユーザ群のことをふぁぼ関係集合と呼ぶ。ユーザは、最大で3200

件までお気に入り登録することができ、それを超えると古いものから順にお気に入り解除されていく。お気に入り機能は、ユーザがツイート1つ1つに対してお気に入り(☆)を付けるかつかないかという評価を下すものであり、ユーザが別ユーザをフォローするか否かの評価を下すフォロー機能とはユーザー一人一人に從属する評価が異なるという性質をもつ(図3)。本稿では、お気に入り機能の星印の数の違いをそのツイートを発信したユーザの評価度合とみなし、ユーザ推薦に用いることとする。

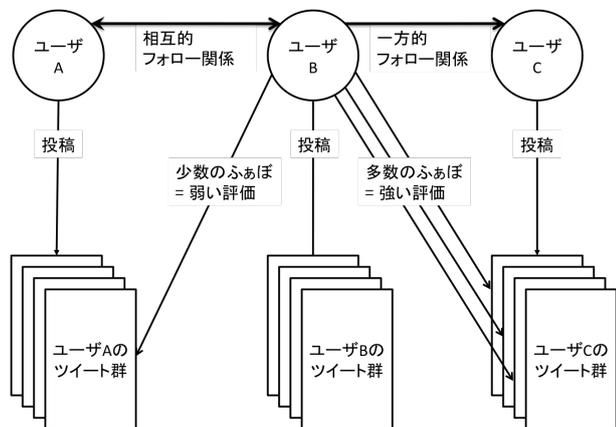


図3 フォロー関係とふぁぼ関係の重み付けの違い

ユーザがお気に入りに登録したツイートは、そのユーザ個別のURLにアクセスすることで一覧することができる。Twitter公式Webサイト(注1)や、外部サイトであるFavstar(注2)、ふぁぼったー(注3)等のサイトがその代表例であり、自分のふぁぼっているツイート、ふぁぼられているツイートを確認することができる(図4)。この時、同じツイートをふぁぼっている他のユーザも確認することが出来る。提案システムは、Favstarのデータを用いて、あるユーザに関するふぁぼ関係を取得する。ふぁぼ関係を取得する範囲は、ユーザの煩雑化、また分析の煩雑化を防ぐため、そのツイートをふぁぼった最新の10ユーザのデータだけに限定する。



図4 Favstarにおけるふぁぼり/ふぁぼられ表示の例

4. 提案システムの対象ユーザ

提案システムの対象ユーザは、Twitterに登録してからある程度の日数が立ち、ある程度以上のユーザからフォローを受けているユーザとする(50名以上が目安)。これは、推薦をする際の基盤データである、ふぁぼ関係やフォロー数、フォロワー数が、有効な推薦のために十分蓄積されるまでの条件として、筆者が経験的に設定したものである。また、この条件を満たしているユーザでも、お気に入り機能を利用したことがなく、かつ、お気に入り機能を使っているユーザがそのユーザの周りにいない場合は、十分有効な推薦が行われないという制限がある。

5. 提案システム

提案システムの概要を説明する。まず、ユーザは、元になる任意のTwitterID(例えば、自分が推薦対象としたい分野で評価が高く、多くの知り合いをもつユーザのTwitterID)をクエリとして提案システムに入力する。そのTwitterIDをもつユーザのふぁぼ関係がFavstarから取得され、推薦の元になるユーザ候補群(ふぁぼ関係集合と呼ぶ。推薦対象としたい分野に関連した様々な分野において評価が高いと予想されるユーザ群に相当)が提示される。次に、ユーザは、ふぁぼ関係集合の中から推薦の元になるユーザをチェックボックスにより絞り込む。これにより、推薦対象としたい分野における、ある特定分野で評価が高く、多くの知り合いをもつと予想されるユーザ群が選択されたことになる。選択されたユーザ群のふぁぼり情報がFavstarから取得され、各ユーザのスコアが計算され、推薦ユーザがランキングされた形でユーザに表示される。図5に提案システムの概要を示す。以下では、各項目を詳しく説明する。

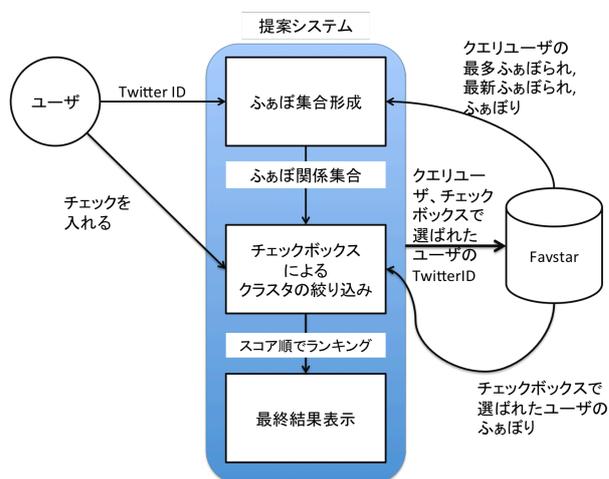


図5 システム概要図

5.1 ユーザにクエリを投げてもらう

まず、ユーザは、ユーザ推薦の起点となるTwitterIDを入力する(図6)。これは、例えば、自分が推薦対象としたい分野で評価が高く、多くの知り合いをもつユーザのTwitterIDなどを想定している。このTwitterIDを基にすることで、ユーザが推薦対象としたい分野でさまざまな評価の高い可能性のあるユー

(注1) : <http://twitter.com/ユーザ ID/favorites>

(注2) : Favstar <http://favstar.fm/>

(注3) : ふぁぼったー <http://favotter.net/>

ザ群がふぁぼ関係集合として得られることが期待できる。一方、この TwitterID のユーザがふぁぼり、ふぁぼられ関係を全く持たない場合、有効な推薦を行うためのふぁぼ関係集合そのものが取得できないことになる。



図 6 クエリ入力画面

5.2 ふぁぼ関係集合の取得

クエリとして入力された TwitterID を用いて、該当ユーザのふぁぼりツイート、ふぁぼられツイートから、ふぁぼ関係集合を生成する。

5.2.1 Favstar へのアクセス

クエリとして入力された TwitterID に該当するユーザの Favstar にアクセスし、そのユーザのふぁぼりツイート、ふぁぼられツイートを取得する (以下、ふぁぼりツイート、ふぁぼられツイートを総称して、ふぁぼツイートと呼ぶこととする)。取得するのは

- 最も現在に近い時間にふぁぼられたツイート群 (以下、 $\{t_r\}$ で表す)
- 最も多くのユーザにふぁぼられているツイート群 (以下、 $\{t_m\}$ で表す)
- クエリのユーザがふぁぼをつけた最新のツイート群 (以下、 $\{t_g\}$ で表す)

の 3 種類である。このふぁぼツイート群の取得においては、それぞれ 20 件までしか取得できないという制限がある。ここでは各 20 件を取得し、推薦に用いる。また、当該ユーザが Favstar にクロールの許可を出してない場合、または、Twitter を公開状態にしていない場合は上記情報を取得できない場合がある。

5.2.2 ふぁぼツイートからのユーザ情報の抽出

取得したふぁぼツイート群のうち、「最も現在に近い時間にふぁぼられたツイート」、「最も多くのユーザにふぁぼられているツイート」の 2 つからは、ふぁぼった側のユーザ情報 (各々 u_r , u_m とする) を、「クエリのユーザがふぁぼをつけた最新のツイート」からはそのツイートを発信したユーザ情報 (u_g とする) を抽出する。ここで抜き出すユーザ情報は、ユーザの TwitterID のみである。すなわち、

$$\begin{aligned} \{u_r | u_r = \text{getTwitterIdGivingFavorite}(t_r), t_r \in \{t_r\}\} \\ \{u_m | u_m = \text{getTwitterIdGivingFavorite}(t_m), t_m \in \{t_m\}\} \\ \{u_g | u_g = \text{getTwitterIdPosting}(t_g), t_g \in \{t_g\}\} \end{aligned}$$

となる。ここで、 $\text{getTwitterIdGivingFavorite}(t)$ はツイート t をふぁぼったユーザの TwitterID を返す関数、 $\text{getTwitterIdPosting}(t)$ はツイート t を発信したユーザの TwitterID を返す関数である。

5.2.3 ふぁぼ関係集合の生成

ふぁぼ関係集合 $userList$ は、5.2.2 節で取得したユーザ情報の和集合をとることで生成される。すなわち、

$$userList = \{u_r\} \cup \{u_m\} \cup \{u_g\}$$

となる。

5.3 推薦ユーザ層の絞り込み

ふぁぼ関係集合 $userList$ に属するユーザの中から、推薦の元になるユーザを 2~3 名程度チェックを付けて選択する (図 7)。ここで図 7 は実際の画面ではなくテスト表示であり、実際はアイコンと ID、それに bio 情報が表示される物である。最終的に推薦されるユーザが持つツイートの話題の傾向や Twitter の使用状況等の属性は、ここでチェックを付けられたユーザと類似している可能性が高い。ここで推薦の元になるユーザを選ぶことにより、最終的に推薦されるユーザは、推薦対象としたい分野において評価が高く、かつ、ユーザが知らない可能性の高いユーザとなるように絞り込まれることにつながると期待される。今図 7 ではゲームについて呟いているユーザと映画について呟いているユーザにチェックを入れているので、最終的なおすすめユーザには、その属性が伝播したと思われるユーザが表示されるという期待が高まるのである。



図 7 推薦の元になるユーザのチェックボックスによる選択の例

5.4 スコアの計算

チェックボックスで選択された、推薦の元になるユーザの Favstar にアクセスし、「ユーザがふぁぼをつけた最新のツイート」 $\{T_g\}$ のみを取得する。他の 2 つのツイート群は受動的なふぁぼられ関係であるために、推薦元ユーザが評価をしているという能動的なふぁぼり関係が得られないため、ここでは取得しない。スコア計算では、推薦の元になる各ユーザ U_g がお気に入りを入れたツイート 20 件 ($=\{T_f\}$) を解析し、各ツイートの発信ユーザ U_f の出現回数をカウントする。全ての推薦元ユーザ $\{U_g\}$ についての各発信ユーザの出現回数の総和を、その発信ユーザ U_f のスコア $Score(U_f)$ とする。すなわち、

$$Score(U_f) = \sum_{U_g \in \{U_g\}} \sum_{T_f \in \{T_f\}} \delta_{U_f, U_p}$$

となる。ここで、

$$\begin{aligned} rcl\{T_f\} &= \text{getFavoriteTweetGivenBy}(U_g) \\ U_p &= \text{getTwitterIdPosting}(T_f) \end{aligned}$$

$$\delta_{i,j} = [i = j]$$

であり、 $\delta_{i,j}$ はクロネッカーのデルタである。得られたスコアが上位の発信ユーザ群 $\{U_f\}$ が、提案システムの最終的な推薦ユーザである。

5.5 推薦ユーザの表示方法

推薦ユーザ群は、5.4 節で計算されるスコアの高い順にソートされて表示される (図 8)。アイコン画像には、該当ユーザの Favstar のページへのリンク、TwitterID には、Twitter 公式のユーザ個別ページへのリンクがそれぞれ設置されている。その直下に配置されている 2 つのボタンのうちの上側ボタンは、おすすめユーザと自分との共通のフォロワーを一覧表示するボタン、下側ボタンは、そのユーザの詳細情報 (フォロワー数、フォロワー数、MeCab [7] のオリジナル辞書を使用したユーザ属性判定の結果等) を表示するボタンである。

おすすめユーザー一覧



主に映画についてつぶやいているユーザー4

bio文が表示されます

このユーザーをフォローしている既知ユーザーを見る

このユーザーの詳細を表示する



主にゲームについてつぶやいているユーザー3

bio文が表示されます

このユーザーをフォローしている既知ユーザーを見る

このユーザーの詳細を表示する



主にゲームについてつぶやいているユーザー4

bio文が表示されます

このユーザーをフォローしている既知ユーザーを見る

このユーザーの詳細を表示する

図 8 推薦ユーザの表示結果

6. 対案システム (Friendly Recommender)

ユーザからユーザへの評価度合を計算できる別の指標にリプライ機能がある。リプライ機能は、ユーザとユーザが会話するための機能であり、「リプライをする=このユーザは親しみやすい」ととらえることにより、ユーザの評価度合が計算できると考えられる。この考えの下、リプライ機能を評価軸とした対案システムを作成した。大まかな手順は Favorite Linker と変わっていない。クエリとなる TwitterID を入力し、最終的なクラスターをチェックボックスによって絞り込み、結果を見る。変わっているのは扱うデータと、そのデータの取得方法、そしてスコアの計算方法である。全体の手順は、お気に入り機能を評価軸とした提案システムと同様である。クエリとなる TwitterID を入力すると、そのふあば関係から、推薦の元になるユーザ候補群 (ふあば関係集合) を生成し、チェックボックスによる絞り込みの結果、最終的な推薦ユーザが得られる。提案システムと異なる

のは、対象データと取得方法、スコア計算方法の内容である。

6.1 データの取得

データの取得には、Twitter の SearchAPI を使用する。SearchAPI は Twitter アカウントの認証が必要ないという理由で選択した。SearchAPI に対し、クエリとして与えられた TwitterID を入力し、該当ユーザの最新 200 件のツイートを取得する。チェックボックスによるユーザの絞り込みで表示されるユーザは、クエリとして入力された TwitterID のユーザが最近 200 件の中でリプライを送ったユーザである。

6.1.1 スコアの計算

チェックボックスで選択された、推薦の元になるユーザのツイート各 200 件分を取得する。各ツイートでリプライを送った先のユーザの出現回数をカウントすることでスコアを計算する。

7. 実験と結果

実験は 2 回行った。研究室で行った実験と、web 上に公開し、ネットユーザーに使用してもらい、感想を貰った実験である。実験概要と結果は実験それぞれについて分けて書くことにした。実験に使うプログラムは同一な物を使用した。

7.1 実験プログラム

実験の際に使用したプログラムでは、Favorite, Friendly などの言葉を使用すると結果に影響を与えると考えられるため、実際にシステムとして使用する画面である図 9 ではなく、図 10 のように、システム 1, 2 という名称で表示を用い、実験を行った。



図 9 実際のトップ画面



図 10 実験時のトップ画面

7.2 おおまかな実験概要

各被験者には、提案システムを 3~4 回程度使用してもらい、結果として得られた推薦ユーザが、自分が期待したユーザとして十分であったか否か、また、推薦ユーザが周りのユーザから評価されているか否かを回答してもらった。被験者自身が考える自分の周辺の評価が高く、自分の知らないユーザと、提案システムが推薦するユーザが類似していれば、提案システムが有

用である可能性を確認できる。また、その後、お気に入り機能を用いた提案システム (システム 1)、リプライ機能を用いた対案システム (システム 2)、Twitter 公式で利用可能なおすすめユーザ推薦 (システム 3) の 3 つからどれが最も満足度が高かったかといった設問等に回答してもらった。

7.3 研究室における実験概要

研究室における、筆者を除いた 10 名のうち、Twitter を使用している 9 人に対し、評価実験を行った。事前にプレゼンテーションをし、操作の仕方を説明。その後実際にシステムを動かしてもらい、感想を書いて貰うという実験手法を取った。

7.3.1 研究室における実験の結果

9 名中 7 名は、普段 Twitter を使用しておらず、ふぁぼ機能を使っていなかったため、提案システムで有効な推薦ユーザを得るための元となるデータが取得できず、結果が何も表示されないこととなってしまった。一方、普段から Twitter を使用し、お気に入り機能やリプライ機能を用いていた 2 名については、チェックボックスによるユーザ絞り込みにより、自分が推薦対象としたい分野に特化したユーザ群が得られやすくなっていること、また、得られたユーザ群は比較的周囲の評価が高いことを確認するに至った。

7.4 web 公開における実験概要

平成 24 年 1 月 23 日 15:00~平成 24 年 1 月 25 日 24:00 分まで実験を行った。被験者は Twitter 上のアカウントで参加者を募集した。web 上に実験サイトと感想投稿用のページ、使用方法を書いたページを用意し、期間中に 15 名から感想を得られた。感想については、以下の様な物を聞いた。

- それぞれのシステムについての点数とコメント
- ユーザインタフェースについて点数とコメント
- またこのシステムを使用したいと思ったか (YES/NO)
- Twitter 公式で公開されているおすすめユーザ推薦機能との比較

7.4.1 web 実験の際の評価ページ

web 実験の際、感想を投稿してもらったページである。枠内はテキストボックスになっており、自由にユーザが記述出来るようになっている。

被験者実験の感想投稿

[感想の書き方が載ってます](#)
[一度御覧ください](#)

被験者様の情報:

```

・Twitter_ID (not必須)
・フォロワー数
・フォロー数
・ツイート数
・ふぁぼり数
・ふぁぼられ数
・日毎にTwitterをしている時間
・日常的にふぁぼを使用しているか
・FovsterやTwi.logといった外部サービスを利用しているか

```

図 11 web 実験の際の評価ページ 1

7.4.2 web 公開における実験の結果

感想用サイトに投稿された数値化出来る設問においては図 11

感想本文(点数について一度御覧ください):

```

→システム1について
・チェックボックスを入れたユーザが評価してそうなユーザが出てきたかどうか
  点数
  コメント

→システム2について
・チェックボックスを入れたユーザと仲が良さそうなユーザが出てきたかどうか
  点数
  コメント

→システム全体について
・ユーザインタフェース(見た目や操作性)について
  点数
  コメント

・Twitter公式のおすすめユーザ推薦と比べてどれが良かったか、何処が良かったか(複数回答可)
  よかった物
 良かった所
  コメント

・また利用したいと思ったか

```

リセット 送信

図 12 web 実験の際の評価ページ 2

に示す結果が得られた。この、Twitter 公式と比べてとは、Twitter 公式のユーザ推薦システムと本研究 (システム 1,2 両方) を比較した場合の本研究の満足度合い。UI とはシステム 1,2 両方のユーザインタフェースの事、釣り合いとは、この本システム (システム 1,2 両システム) を使用した際のユーザの労力と出てくるユーザとの釣り合い度合いの事である。他の設問の答えとして、再度このシステムを使用したいと思ったかについては 11 名が YES、3 名が NO、1 名が保留であった。良いコメントを数点以下に要約し、記載する。

- フォロワーの多いアカウント (公式や有名人) ばかり出てきて Twitter をコミュニケーションツールとして使用している私のようなユーザには役不足な Twitter 公式のおすすめ推薦よりも自分フォロワーの近接ユーザが出てくる
- 自分の ID 入力後、チェックボックスでの指定で更にユーザーを絞れる
- 共通の趣味などでよく絞れてフォローしやすい状態ができています
- 公式と違い自分が絡んでる人がよく絡んでる人を表示してくれるので、自分も話題に入っていくやすい、絡みやすい人が出る

悪いコメントを数点以下に記す。

- バグ、またはエラーが多い
- API エラーでアイコンや自己紹介が表示されない
- チェックを入れる画面で知らない人が出現し、何を基準にチェックすれば良いのか判らなかった
- 最終結果表示画面の右側の余白が寂しい
- ブロックしているユーザが出てきた

8. 実験に対する考察

図 11 に示されている不満点のほぼ全てが API 制限についての事だった。API 制限とは、時間あたりに Twitter に対しデータを取得できる回数が制限を超えてしまい、データが取得できなくなる状態の事である。本システムは、本システム 1 連の流れ (クエリを入力、チェックボックスを入れ送信、結果表示) を

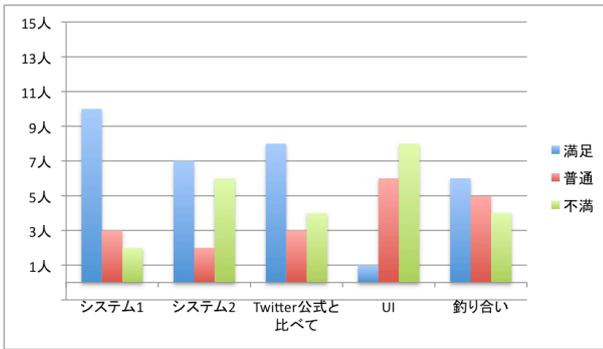


図 13 web 公開実験の結果

2~3 回繰り返すと API 制限になってしまい、正常にデータが取得できず、ユーザのアイコンや自己紹介文、またフォローしているユーザとフォローしていないユーザの区別がつかなくなり、最終結果に既にフォローしているユーザが出てくるという問題を抱えているが、それが見事に表面化されたという結果であった。

評価されている部分については、Twitter の一部分で行われている行為である「空中リプライ」の宛先であろうユーザが手軽に発掘出来るという本研究の狙いどころが表面化した結果であると判断した。空中リプライとは、リプライ先を明記せずに直接 TL の誰かに当たった発言を呟くと言う行為であり、基本的に仲の良いユーザ同士で行われる行為である。この空中リプライを主に使用しているユーザらを特定するにはフォロー関係を重視した推薦では全フォロワーの重さが均一である点から発見がしづらく、そういったユーザを本研究が見つけれられるということで、その点が有用であることが証明された。

Twitter 公式と比べて本研究の満足度が高い理由としては、チェックボックスにより自分の望んだユーザ層のユーザが出てくる事、データを取得するユーザをある程度任意に選べることだった。この点に対する不満点としては、やはり TwitterAPI 制限による UI の見づらさが多数上がっていた。

9. 結 論

本稿では、Twitter のお気に入り機能に着目し、これに基づくユーザ毎の評価を考慮した推薦システムを提案した。これにより、フォロー関係のみでは発見が困難だったある局所的なコミュニティにおけるユーザ間で評価の高いユーザを発見しやすくなる事が期待される。予備実験の結果、また寄せられたコメント等から見るに、Twitter をコミュニケーションツールとして使用しているユーザに対して、自分が推薦対象としたい分野に特化したユーザ群が得られやすくなっていること、また、得られたユーザ群は比較的周囲の評価が高いことが確認され、想定したユーザを見つけやすいという感触が得られた。対案としたリプライ機能を用いた Twitter ユーザ推薦に関しても、同様の事がわかった。お気に入り機能よりもリプライ機能の方の評価が低い理由としては、リプライは基本フォローしあっているユーザ同士の元行われる物であり、リプライしあっているユーザ間の関係を抽出している際、ユーザが API 制限にひっかか

ると、お気に入り機能よりもフォローしているユーザが出てきやすいという現象に起因していると考えられる。しかし、そういったエラーがあるにも関わらず、良いコメントが来たことで、自分が推薦対象としたい分野に特化したユーザ群が得られやすくなっていること、また、得られたユーザ群は比較的周囲の評価が高いことが確認され、想定したユーザを見つけやすいという感触が得られた。

その他の感触として、本研究をする中で、また筆者の普段の Twitter 上の経験からの感触ではあるが、ふぁぼ機能を全く使っていない Twitter ユーザはかなり多く存在すると推測され、日常的にふぁぼ機能を使用しているユーザはまだまだほんの一部であると感じている。今後、ふぁぼ機能の利用割合を考慮した検討も行っていきたい。

最後に、本研究の問題点とその解決法を示す。

- システムを 1 回動かす (クエリ入力から推薦ユーザの結果が返ってくるまで) のにかかる時間は、待ち時間を含め 1~2 分程度であるが、Favstar や Twitter サーバといったデータ取得先の外部サイトが混み合っていると、待ち時間だけで 1 分以上かかることもある。ローカルに自前のふぁぼ DB を持つことを検討していきたい。
- クエリの入力画面や、結果を表示画面のユーザインタフェースの有用性が確認できていない。今後、評価を進める必要がある。また、今回の実験で得られた意見を元に改良も行う必要がある。
- ユーザのスコア計算が単純な出現回数であるので、ページランクや HITS アルゴリズムなどを応用したスコア計算も検討の余地があると考えられる。ただ、提案システムの推薦結果の妥当性やユーザインタフェース有用性に関する比較評価を先に進めいきたい。
- ふぁぼを使用していない、リプライを使用していないユーザに対して、推薦ユーザを提示できない点は改善を検討したい。
- 不満意見の出現元である API 制限に関しては、Twitter の OAuth 認証を用いる事による制限の緩和を行う事で対処を試みる。

文 献

- [1] Marshall Kirkpatrick, Twitter Confirms It Has Passed 200 Million Accounts, 70 % of Traffic Now International http://www.readwriteweb.com/archives/twitter_confirms_it_has_passed_200_million_account.php
- [2] Twitter/おすすめユーザー/あなたへのおすすめ https://twitter.com/#!/who_to_follow/suggestions
- [3] Twitter/○○さんに似ているユーザー https://twitter.com/#!/similar_to/○○のTwitterID
- [4] 康大樹, 島田論, 関洋平, 佐藤哲司: 属性伝搬モデルを用いたマイクロブログのフォロー先推薦法, DEIM Forum 2011, A1-3, 2011
- [5] 田中淳史, 田島敬史: twitter のツイートに関する分類手法の提案, DEIM Forum 2010, A5-4, 2010
- [6] The Best of Twitter. See your most popular tweets. <http://ja.favstar.fm/>
- [7] 京都大学情報学研究所-日本電信電話株式会社コミュニケーション科学基礎研究所共同研究ユニットプロジェクト, MeCab: Yet Another Part-of-Speech and Morphological Analyzer <http://mecab.sourceforge.net/>