

# 平成24年度修士論文

## Twitterにおける リツイート情報の重ね合わせによる 情報提供ユーザ推薦

学籍番号 1131025

氏名 太田 侑介

情報・通信工学専攻 コンピュータサイエンス  
コース

指導教員 寺田 実 准教授

副指導教員 角田 博保 准教授

提出日 2013年 1月25日

# 概要

## 目的

近年, 様々なソーシャルネットワーキングサービス (SNS) が出現し始めた. ユーザは SNS 上で自分の考えや興味ある情報を発信することで, 現実の世界では出会えなかったようなネット上の友人を得ることができる. しかし, 他ユーザとのつながりが増えるほど, ユーザに与えられる情報は膨大になっていく. 特に Twitter では, タイムライン上にあるツイート数が多くなり, ユーザの興味あるツイートが雑多なツイートに埋もれてしまう. 本研究の目的は Twitter において, ユーザに有益な情報を与えることである. そのため, 本研究ではリツイートについて着目した. リツイートされたツイートは面白い, 興味深いと判断されたツイートである. 本研究では対象ユーザにとって有益なツイートをよくリツイートしてくれるユーザを推薦する.

## 方法

対象ユーザとリツイート共有回数が多いユーザが推薦ユーザとなる. 提案するシステムは, リアルタイムに Twitter のタイムラインを監視し, リツイートが行われるたびに推薦ユーザを更新する. ユーザはリツイートを行うだけで, 自動的に推薦ユーザを更新することができる. また, Twitter 内のリスト機能やシステムが提供するリツイート Viewer を使うことでユーザは推薦ユーザから与えられる情報を確認できる.

## 結論

評価実験の結果, 確かに推薦ユーザが与える情報にはユーザの興味ある内容を含みやすいことが分かった. そのため, 本システムが推薦したユーザからの情報を見ることで, ユーザが得る興味あるツイート率はよくなった.

# 目次

<b>第 1 章 序論</b>	<b>6</b>
1.1 背景	6
1.2 問題点	6
1.3 着眼点	6
1.4 目的	6
1.5 論文構成	7
<b>第 2 章 Twitter</b>	<b>8</b>
2.1 概要	8
2.2 リツイート	8
2.2.1 非公式リツイート	9
2.2.2 公式リツイート	10
2.2.3 リツイートの長所	11
<b>第 3 章 ユーザの興味判別</b>	<b>13</b>
3.1 フォロー関係	13
3.2 ツイート内容分析	13
3.3 リツイートユーザ	14
<b>第 4 章 先行研究</b>	<b>16</b>
4.1 概要	16
4.2 リツイート経路木	16
4.2.1 伝搬経路推定	16
4.2.2 伝搬経路の可視化	17
4.3 オーバーラップグラフ	17
4.4 評価	20
4.4.1 ツイート及びリツイート分類	20
4.4.2 カテゴリ特徴ベクトル	21
4.4.3 結果	22
<b>第 5 章 提案システム</b>	<b>23</b>
5.1 概要	23
5.2 情報提示方法	23
5.2.1 リスト管理システム	23
5.2.2 リツイート専用 Viewer	23
5.3 特徴	24
5.3.1 自動リスト管理機構	24

5.3.2	未知ユーザの発見 . . . . .	24
<b>第 6 章</b>	<b>実装</b>	<b>25</b>
6.1	開発環境 . . . . .	25
6.2	システム運用 . . . . .	25
6.3	推薦ユーザ導出アルゴリズム . . . . .	25
6.4	初期状態生成 . . . . .	27
6.5	OAuth 認証 . . . . .	27
<b>第 7 章</b>	<b>評価実験</b>	<b>29</b>
7.1	概要 . . . . .	29
7.2	手法 . . . . .	29
7.3	データセット . . . . .	29
7.4	結果 . . . . .	30
<b>第 8 章</b>	<b>重みづけによる推薦ユーザ変化シミュレーション</b>	<b>32</b>
8.1	概要 . . . . .	32
8.2	データセット . . . . .	32
8.3	手法 . . . . .	32
8.4	結果 . . . . .	32
<b>第 9 章</b>	<b>結論</b>	<b>34</b>
9.1	結論 . . . . .	34
9.2	今後の課題 . . . . .	34
9.2.1	リアルタイム性のあるユーザの興味に対する対処 . . . . .	34
	<b>参考文献</b>	<b>36</b>

## 目次

2.1	リツイート例 . . . . .	8
2.2	非公式リツイート例 . . . . .	9
2.3	公式リツイート例 . . . . .	10
3.1	TwitterGraph . . . . .	14
4.1	リツイート伝搬経路木 . . . . .	16
4.2	オーバーラップグラフ構成例 . . . . .	18
4.3	オーバーラップグラフ . . . . .	19
4.4	評価手順1 . . . . .	20
4.5	評価手順2 . . . . .	21
5.1	リツイート専用 Viewer . . . . .	24
6.1	システム図 . . . . .	26
6.2	OAuth 認証例 . . . . .	28
7.1	有益率結果 . . . . .	30

# 表 目 次

3.1	リツイートリスト例 . . . . .	14
4.1	評価結果 . . . . .	22
6.1	使用した API . . . . .	25
7.1	評価結果 . . . . .	30
7.2	タイムラインにおける評価結果 . . . . .	31
8.1	シミュレーション結果 (推薦ユーザリスト, スコア) . . . . .	33

# 第1章 序論

## 1.1 背景

近年では、Twitter<sup>1</sup>や Facebook<sup>2</sup>などの多くのソーシャルネットワーキングサービス (SNS) が出現し始めた。ユーザは SNS を友人とのコミュニケーションツールや情報収集ツールとして利用している。また、ユーザは SNS 上で自分の考えや興味ある情報を発信することで、現実の世界では出会えなかったようなネット上の友人を得ることができる。SNS の中でも Twitter は、ユーザはツイートという短いメッセージを用いて、気軽に情報を発信することが可能である。ユーザは他のユーザをフォローすることで、そのユーザのツイートが自分のタイムライン上に表示される。そのため、多数のユーザをフォローするほど、タイムラインに流れてくるツイート数は増大する。

## 1.2 問題点

Twitter を使っていく上で、ユーザはコミュニティ関係を広げるためや、純粋に情報収集のためにフォローするアカウントが増える。すると、当然ながらタイムラインに流れてくるツイート数も膨大になっていく。ここで、フォローしたユーザは常に自分にとって興味ある内容をツイートするとは限らないため、雑多で不要な情報も増えてしまう。ある程度のツイート数を超えてくると、ユーザは全てのツイートを見るのは困難になってしまう。その結果、自分にとって有益な情報が不要な情報に埋もれてしまい、把握できない可能性が発生してしまう。

## 1.3 着眼点

そこで本研究はリツイートという機能に着目した。リツイートされたツイートはユーザによって面白い、興味深いと判断されたツイートである。そのため、自分の興味あるツイートをよくリツイートしてくれるユーザを重点的に監視することで、自分に有益な情報を得やすくと考えた。

## 1.4 目的

本研究の目的はユーザに有益なツイートをよくリツイートしてくれるユーザを推薦することである。リアルタイムに Twitter のタイムラインを監視し、リツイートが行われるたびに推薦ユーザを更新する。また、Twitter 内のリスト機能やシステムが提供するリツイート Viewer を使うことでユーザは推薦ユーザから与えられる情報を確認できる。

---

<sup>1</sup><http://twitter.com/>

<sup>2</sup><http://www.facebook.com/>

## 1.5 論文構成

本論文の構成は以下の通りである。

第2章では, Twitter について述べる.

第3章では, ユーザの興味発見方法について述べる.

第4章では, 先行研究であるオーバーラップグラフについて述べる.

第5章では, 提案システムについて述べる.

第6章では, 提案システムの実装方法について述べる.

第7章では, 評価方法について述べる.

第8章では, ユーザ推薦に重みづけを考慮したシミュレーションを行った結果を述べる.

第9章では, 結論について述べる.



## 第2章 Twitter

### 2.1 概要

Twitter はツイートと呼ばれる 140 字のメッセージからなるマイクロブログである。ツイートは非常に短い文章であるため、ユーザはいつでも思ったことを気軽にツイートすることができる。また、ユーザは他のユーザをフォローすることでそのユーザのツイートを見ることが出来る。フォローしているユーザ達、されているユーザ達はそれぞれフォロワー、フォローと呼ばれる。ユーザのタイムラインには自分のツイートとフォロワーのツイートが表示される。Twitter には、ツイートに対する返答をするリプライ機能や気に入ったツイートに目印をつけるお気に入り機能、ツイートにタグをつけるハッシュタグ機能など様々な機能がある。その中でも本研究が着目したリツイートについて説明をする。

### 2.2 リツイート

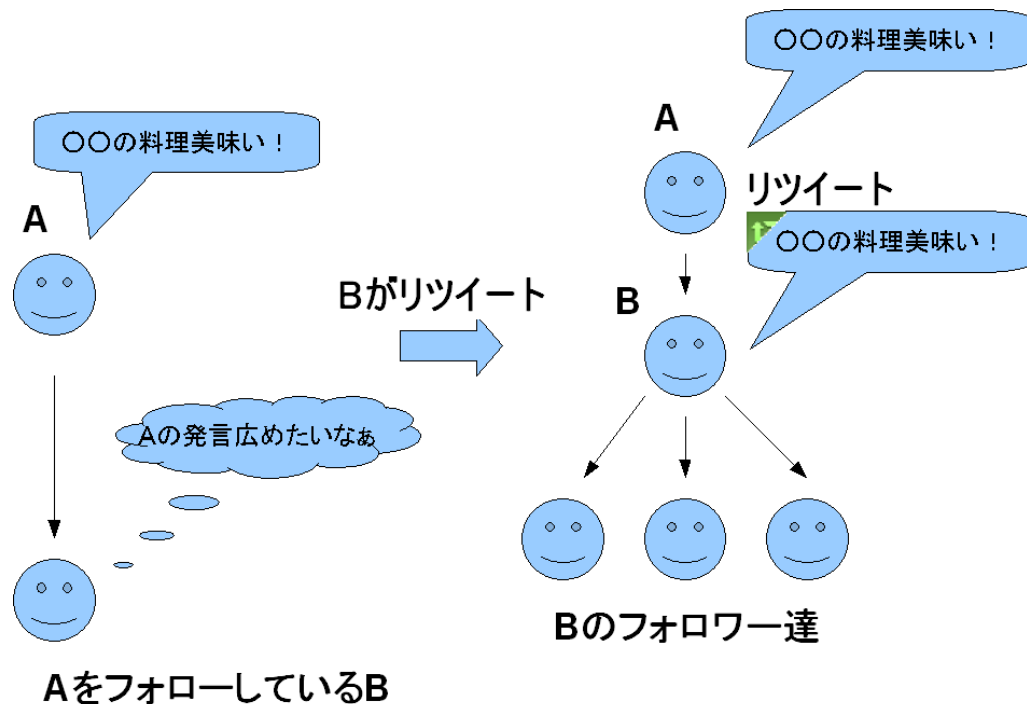


図 2.1: リツイート例

本研究では Twitter のリツイートという機能に着目した。あるユーザが他人のツイートを自分をフォローしているユーザにも提示する機能がリツイートである。リツイートは図 2.1 のように、他人のツイートを自

分のツイートとして再投稿することで、自分のフォロワーにもその内容を広めることができる。また、それを見たユーザがさらにリツイートをしていくほど、多数のユーザにツイートが拡散され、元のユーザから関係が遠いユーザにもツイートが共有される。

### 2.2.1 非公式リツイート



図 2.2: 非公式リツイート例

リツイートには Twitter 側から機能として提供されている公式リツイートと図 2.2 のように、ユーザがメッセージを付けて疑似的にリツイートの形にした非公式リツイートの 2 種類が存在している。Twitter では、もともとユーザ間の中で非公式リツイートの形で興味深い情報を多くのユーザに知らせていた。だが、非公式リツイートには以下のような問題点が存在する。

- 帰属について混乱
- ツイートの正確性
- 冗長性と騒音
- 追跡の不備

#### 帰属についての混乱

通常のツイートはアイコン、ユーザ名、テキスト本文という要素で成り立っている。そのため、ツイートしたユーザははっきりとわかる。しかし、非公式リツイートの場合、アイコンはリツイートユーザのものであったり、リツイート元ユーザ名は本文中にしか書かれないため、あたかもリツイートしたユーザがその本文を書いたかのように見間違えてしまうことがある。大量のツイートがある中で、一目でリツイート内容が誰のものであるか分からないのは問題である。

### ツイートの正確性

非公式リツイートを行う場合、元の文にリツイートしたユーザのコメントを載せる形になる。しかし、元の文が改変可能であるため、リツイート内容がオリジナルのものでない可能性もある。このことは、スパムツイートの原因になりうる。例えば有名人のツイートを改変してリツイートすることで、その有名人に悪影響を及ぼすことができってしまう。

### 冗長性と騒音

あるユーザのフォロワー達が一つのツイートに対して非公式リツイートを一斉に行った場合、ユーザには同じ文章についてリツイートを複数受け取ることになる。ユーザにとってこのことは冗長な情報である。リツイートについてのコメントを探すときに、それら情報は騒音となる。

### 追跡の不備

非公式リツイートは本文中にリツイートユーザの情報を載せるため、リツイートが多くされたツイートの場合、ツイートの文字数制限から過去のリツイートユーザの情報は必然的に消えることになる。このことはリツイートの情報をマイニングすることを困難とする。

## 2.2.2 公式リツイート



図 2.3: 公式リツイート例

## 長所

Twitter は非公式リツイートの欠点を補強するために公式にリツイート機能を提供した。図 2.3 は公式リツイートの例である。あやふやであった、非公式リツイートを構造化することで、リツイートに関わったユーザの情報やリツイート自身の情報などが正しく保持される。公式リツイートはリツイートしたユーザのリストを保存しており、何度リツイートされようともリツイートしたユーザの情報が消失することはない。また、リツイート本文は変更できないため、内容の正確性は保持される。

## 短所

だが、構造化したことで、公式リツイートにはいくつかの欠点も生まれた。まず、リツイート本文を改変できなくなったことから、リツイートに対するコメントをし辛くなったことである。ユーザが公式リツイートに対してコメントをする場合は、通常のリツイートやリツイート元ユーザに対してリプライをするなどの方法しかコメントができない。次に、リツイートユーザリストは取得できるが、誰からリツイートを行ったかという情報はわからないということである。もし、リツイートをマイニングに利用する際、リツイートの伝搬経路が必要になった場合は、リツイートした時刻などから推測するなどの方法を取らなければならない(4.2.1 参照)。

このように、公式リツイートは必ずしもユーザにとって良いことばかりというわけではない。そのため、ユーザはリツイートする際に、公式、非公式のどちらが適切であるか判断して利用する必要がある。

本研究では、リツイートを行ったユーザの情報が重要となってくるため、公式リツイートについてのみ扱う。

### 2.2.3 リツイートの長所

リツイートには以下のような長所がある。

- ユーザによるフィルタリング
- 同一興味ユーザ集団の発見
- アクティブユーザの発見
- 関係の遠いユーザの発見
- 話題の抽出

#### 同一興味ユーザ集団の発見

Boyd らの研究 [2] によると、ユーザはあるツイートが面白い、興味深い、他のユーザに伝えたい、という理由でリツイートを行う。つまり、同じリツイートをしているユーザ達はそのツイート内容に対して同一の興味を持っていると考えられる。そのため、リツイートをユーザ間の興味尺度を図るバロメータとして利用することができる。

### ユーザによるフィルタリング

多くのユーザにリツイートされるようなツイートは、前項の理由より、より多くのユーザに受け入れられ、面白いと思われた良いツイートである。ユーザはリツイートを行うことで無意識的にツイートが良いものであるかどうかを判断している。つまり、リツイートという行為はツイートをユーザの感性というある種のフィルタを通すことである。そのため、リツイートされたツイートは通常のツイートよりも良いものであると判断できる。

### アクティブユーザの発見

リツイートを行うユーザは、自分の考えをツイートするだけでなく、他のユーザのツイートについて興味を持ち、それを広めようとしている。つまり、積極的に Twitter を使っていると考えられる。Twitter を活発に利用して、情報を拡散させようとしているユーザからは様々な新鮮な情報を得ることができる。

### 関係の遠いユーザの発見

リツイートには情報の拡散性があるため、よりリツイートされているツイートほど、リツイートを行った関係の遠いユーザと出会うやすくなる。本来ならば、新たなユーザをフォローしようとする場合、自分と関係が近いユーザを選択しやすい。しかし、リツイート情報を利用することで、自分と関係が遠いユーザを発見しやすいこととなる。リツイート内容について同一の興味を持っているため、ランダムに探すよりも効率が良い。

### 話題の抽出

Suh ら [4] の調査では、フォロワー数が多かったり、Twitter アカウントが作られた時期が古いユーザのツイートはリツイートされやすいという結果が出ている。このようなユーザはツイート内容の話題界限では著名なユーザであるといえる。すなわち、その話題について興味あるユーザにとって、著名ユーザのツイートは重要な情報である。よりリツイートされるほど、その重要度は高いと考えられる。リツイートは興味ある話題についてのツイートの重要度を定める指標となりうる。

## 第3章 ユーザの興味判別

自分に有益な情報を与えるユーザを探すためには、ユーザの興味を判別する必要がある。Twitterにおいてユーザの興味判断に使える情報は様々なものがある。

### 3.1 フォロー関係

ユーザが他のユーザをフォローする場合、フォローするユーザは自分に興味ある内容をツイートするユーザであると考えられる。そのため、フォロー関係からユーザのクラスタリングを行った場合、同一クラスタ内のユーザは興味が似ていると判断できる。

フォロー関係からユーザのクラスタリングを行う場合、協調フィルタリングという手法が有効である。同じ興味を持っているユーザは同じようなものを好むとするのが協調フィルタリングの考え方である。協調フィルタリングは主に Amazon などイーコマースサイトにおいて、商品の推薦システムに利用されている。例えば、あるユーザ *A* がいたとき、ユーザ *A* と興味が似ているユーザ *B* が買っている商品をユーザ *A* にも推薦する。このことから、同じような商品を買っているユーザは同一の興味を持っていると推測することができる。Twitter について利用すると、同じようなユーザをフォローしているユーザ達は同一のクラスタであると考えることができる。Hannon ら [5] はツイートや Twitter におけるソーシャルグラフからフォロワー対象ユーザ発見を促した。彼らのソーシャルグラフを利用した手法では、ユーザのフォロワー ID、フォロワー ID、その両方の ID という 3 種類をそれぞれユーザのプロフィールと見立てて、協調フィルタリングを用いてユーザ推薦を行った。また、フォロー関係を可視化したものとして TwitterGraph<sup>3</sup> を挙げる。TwitterGraph は、表示されているユーザ *A* をクリックすると、ユーザ *A* のフォロワーの一部が表示される。もし、もともと表示されていた別のユーザ *B* と新たに表示されたユーザ *A* のフォロワーとの間にフォロー関係がある場合、新たに線が繋がれる。その結果、ユーザ *A* とユーザ *B* との共通のフォロワーが浮かび上がる。この共通のフォロワーを一つのクラスタとして見ることで、ユーザのクラスタリングをすることができる。

このようにフォロー関係を利用すると、興味が類似し、関係が近いユーザを探すことが容易となる。

### 3.2 ツイート内容分析

ユーザのツイート内容には、そのユーザの興味ある内容を含んでいることが多い。そのため、ユーザのツイート内容を分析することでユーザの興味を見つけることができる。Michelson ら [6] は、Wikipedia を知識ベースとしてツイート内容のカテゴリ化を行い、ユーザの興味を発見した。彼らは、ツイート内容から重要なワードをいくつか発見し、それらから上位のカテゴリを決定する。そして、ユーザの複数のツイートに対してカテゴリ化を行い、最終的にユーザの興味を推測した。

また、ツイートは 140 字の非常に短いメッセージであるため、通常の文章解析手法の場合良い結果が出にくいことが多い。西田ら [1] は形態素解析に依存せず、学習対象の変化に素早く追従可能なアルゴリズムと

<sup>3</sup><http://twittergraph.wetcradle.com/>

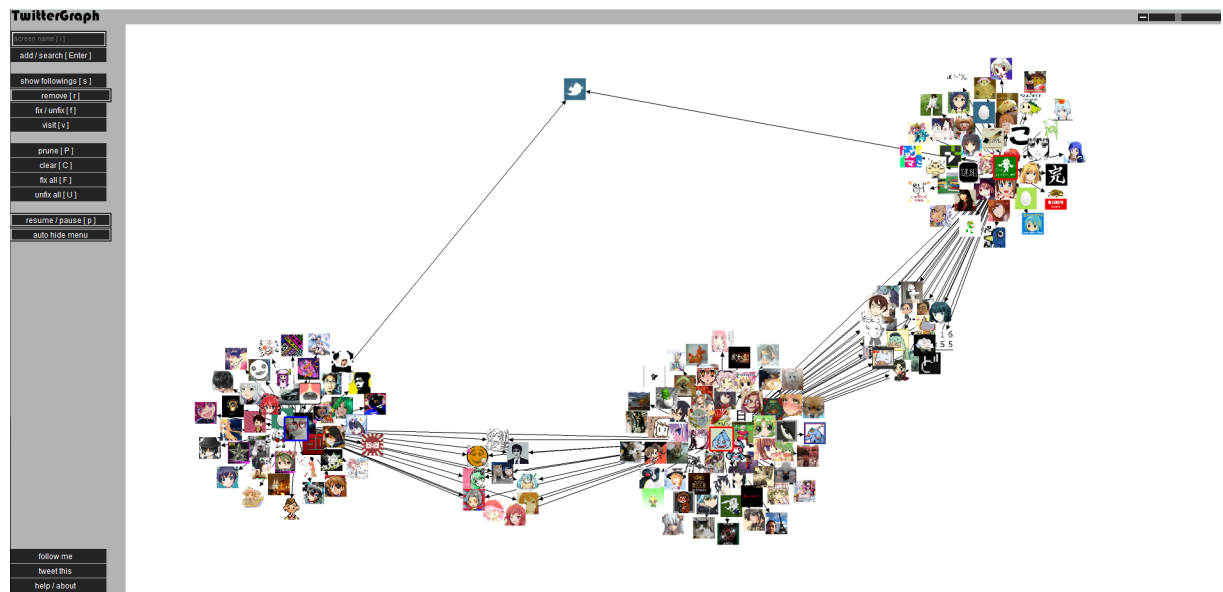


図 3.1: TwitterGraph

して、データ圧縮によるテキストの圧縮され易さを応用した分類方法を提案した。彼らはツイートに着目する話題に関するツイートの集合 (話題モデル) とそれ以外のツイートの集合 (比較モデル) の両方を基に圧縮したとき、どちらのモデルを基にしたほうが圧縮しやすいかによって分類を行った。もし、話題モデルのほうが圧縮しやすい場合はその話題に関する可能性が高いとみなした。このようにツイート分析には、ツイートの特徴を利用する手法が必要となる。

### 3.3 リツイートユーザ

表 3.1: リツイートリスト例

User	Retweet A	Retweet B	Retweet C	Retweet D	Retweet E
W	○	×	○	○	○
X	×	○	○	○	×
Y	×	×	○	○	○
Z	○	○	×	×	×

2章で述べたとおり、リツイートしたユーザ達はその内容について興味を持っていると考えることができる。つまり、ユーザの行った複数のリツイート情報をユーザの興味全体であるとみることができる。

また、協調フィルタリングの考えから興味似たユーザを見つけることが可能である。表 3.1 は複数のリツイートをそれぞれ行ったかどうか示した例である。例えば、ユーザ W はリツイート A, C, D, E に関してリツイートを行った。リツイートをユーザの興味指標としてみる場合、ユーザ W はリツイート B 以外の内容に興味を持っていると考えることができる。また、ユーザ Y はリツイート C, D, E に関してリツイートを行い、それぞれに興味を持っている。この結果から、ユーザ W とユーザ Y の興味は類似していると考え

ことが可能である。このように、リツイートを協調フィルタリングの考えに利用することで、興味類似ユーザを見つけることができる。

また、ユーザの興味内容まで知りたい場合は前節のようにツイート内容を評価する必要がある。しかし、本研究の目的においては同一の興味を持つユーザを探すことが重要であり、その内容は考慮する必要がない。そのため、リツイートを多く共有しているかどうかを見るだけで興味が類似したユーザを発見することができるリツイート情報は本研究にとって有用であると考えられる。また、内容を解析する手間がないため、ツイート本文の言語によらず、興味が似ているユーザを発見することが可能となる。



## 第4章 先行研究

### 4.1 概要

先行研究として本論文著者が作成したオーバーラップグラフ [14] について述べる。2章に論述したように、リツイートをしているユーザ達は同一の興味を持っている。すなわち、リツイートを共有するほど、興味がより類似していると考えられる。先行研究では、まず特定のユーザ(センターユーザ)の行ったリツイートを収集し、それらの伝搬経路を推定し、経路木として可視化した。そして、それら経路木を重ね合わせて、複数回登場したユーザを可視化したオーバーラップグラフと呼ばれるものを作成し、フォロー対象者発見を促した。この先行研究の考え方である、リツイート情報の重ね合わせという観点を本研究にも利用する。

### 4.2 リツイート経路木

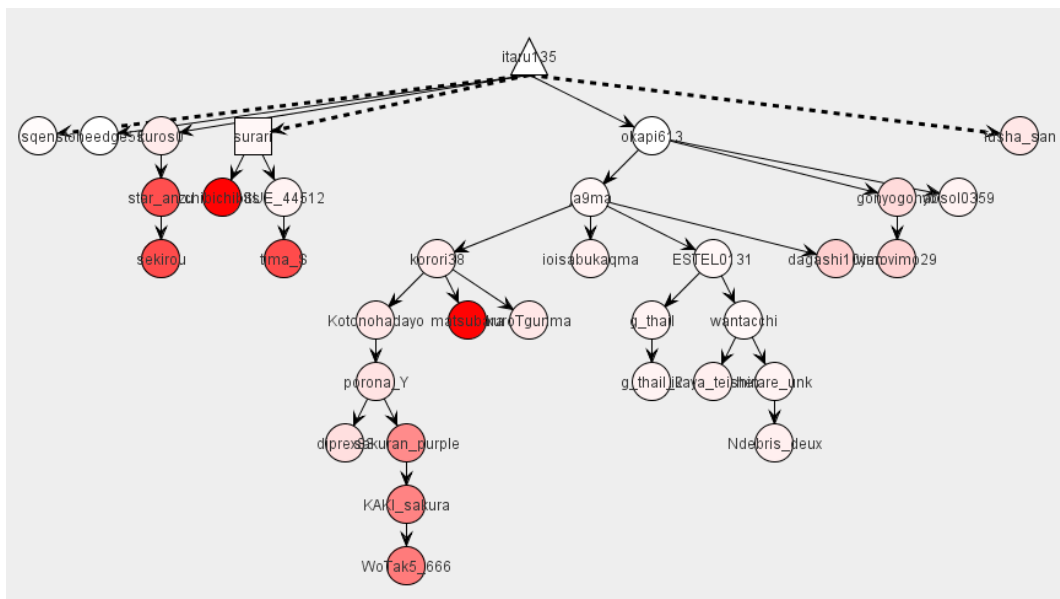


図 4.1: リツイート伝搬経路木

#### 4.2.1 伝搬経路推定

Twitter から提供される公式リツイートの情報には、ユーザがリツイートした時刻は含まれているが、誰からリツイートしたかという情報は含まれていない。そのため、リツイートした時刻から伝搬経路を推定し

た。ユーザは自分のタイムラインに流れてきたリツイート (or ツイート) か検索などで見つけたリツイート (or ツイート) に対してリツイートを行う。前者の場合、タイムラインに表示されるのはフォロワーのものに限られる。そのため、自分より早くリツイートを行ったフォロワーからリツイートを行ったと推定した。もし、該当するユーザがいなかった場合、リツイート元のユーザから直にリツイートを行ったとした。また、後者の場合も同様にリツイート元からリツイートを行ったとした。

#### 4.2.2 伝搬経路の可視化

図 4.1 が実際に伝搬経路木を可視化した例である。三角のノードがリツイート元 (ルートユーザ) であり、四角のノードがセンターユーザである。リツイートは矢印の方向に伝搬される。ノードの色がリツイートした時刻を示しており、色が濃いほど時刻が後になっている。また、Twitter では、リツイートを行った後に、そのリツイートを解除することができる。伝搬経路中のユーザがリツイートをやめた場合、そのユーザ以降にリツイートしたユーザ群の伝搬経路を辿ることができなくなる。そういった場合はルートユーザから直接リツイートをしたと推測し、エッジを点線で表すことで差別化をした。

リツイートの伝搬経路を可視化することで、リツイートにおけるユーザ間の関係を知ることができる。

### 4.3 オーバーラップグラフ

オーバーラップグラフは、作成した複数のリツイート伝搬経路木について、二回以上登場しているユーザを集め、そのユーザからリツイート元までの経路を重ね合わせたものとなっている。図 4.2 はオーバーラップグラフの構成例である。リツイート伝搬経路木 A, B, C を重ね合わせた結果、二回以上登場したユーザとそのユーザからルートユーザまでの経路途中に出てきたユーザ、それぞれのリツイートのルートユーザがオーバーラップグラフには存在する。

図 4.3 は実際に作成したオーバーラップグラフ例である。オーバーラップグラフ上の要素は以下のようになっている。

- 四角ノード：センターユーザ
- 三角ノード：ルートユーザ
- ノード色の濃さ：リツイート共有回数
- エッジの方向：リツイート伝搬方向
- エッジの太さ：パスの個数

オーバーラップグラフからはどのユーザがセンターユーザと興味がより類似していて、そのユーザとの関係がどうなっているかということを理解しやすい。特に図 4.3 中ユーザ A のようなセンターユーザとは関係が遠いが、興味が類似しているユーザを発見することができる。このユーザは、自分の知らないところで、複数の同じツイートに対して自分と同様にリツイートを行っているユーザである。個々のリツイート経路を可視化してもそのユーザはただのユーザと扱われていた。だが、リツイートの情報を重ね合わせたことがこのユーザの発見に繋がっている。よって、オーバーラップグラフではこのような潜在的に自分と同じ興味を持っているユーザをグラフ表示することで直感的に発見することに優れていると言える。

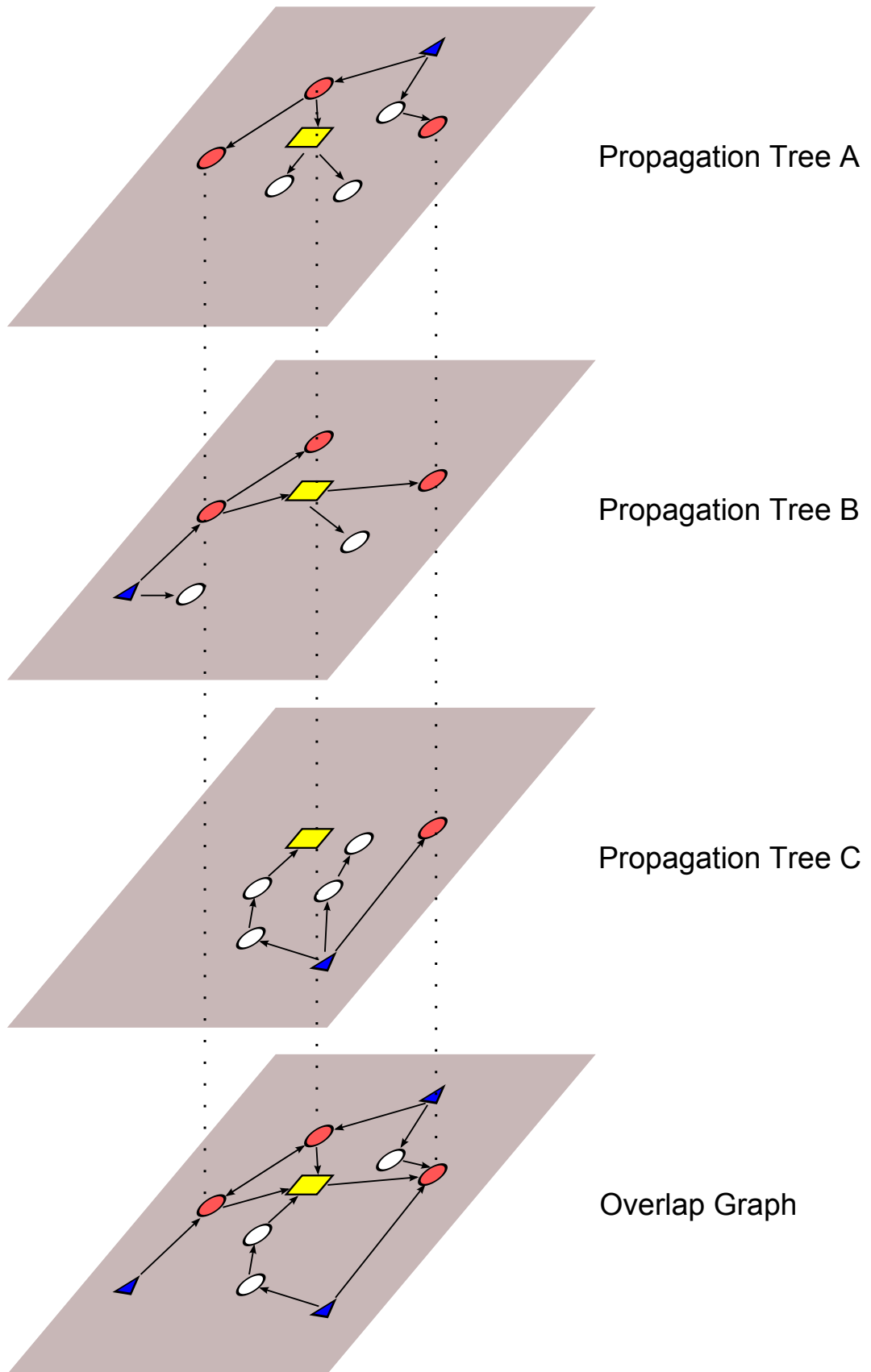


図 4.2: オーバーラップグラフ構成例



## 4.4 評価

オーバーラップグラフから見つかったユーザ達が本当にセンターユーザと同じ興味を持っているかを評価する必要がある。先行研究では、発見したユーザ群 (Highest ユーザ) とセンターユーザと 1 件しかリツイートをしていないユーザ群 (Lowest ユーザ) を以下の要素から評価し、比較した。

- リツイート数
- センターユーザの興味あるリツイート数
- センターユーザとのコサイン類似度

また、評価に使ったデータセットは以下のとおりである。

- センターユーザのリツイート 100 件
- 3 日分の Highest ユーザ, Lowest ユーザのリツイート含むツイート

### 4.4.1 ツイート及びリツイート分類

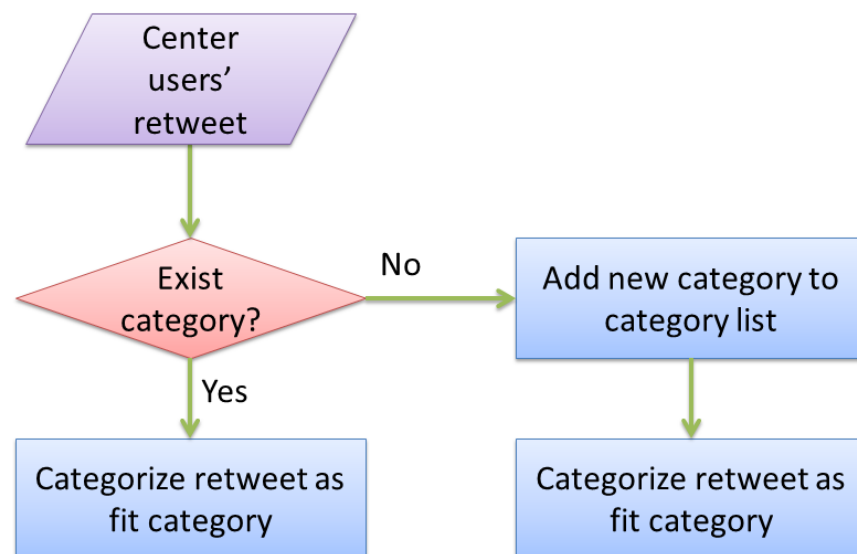


図 4.4: 評価手順 1

ツイート分類の手順として、まずセンターユーザは図 4.4 のように、自分のリツイートを自身の手で分類する。このとき、すでに分類したカテゴリが存在するならそのカテゴリに、ないなら新たにカテゴリを作成

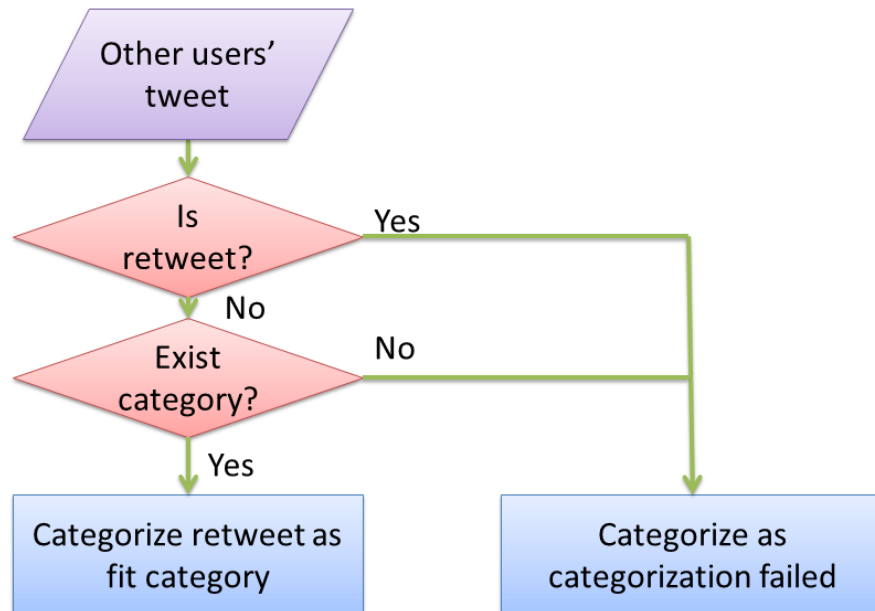


図 4.5: 評価手順 2

して分類する。ここで、分類された  $n$  個のカテゴリを  $CAT_1 \sim CAT_n$  と表す。次にセンターユーザは図 4.5 のように、Highest ユーザのツイートの中からリツイートのみを  $CAT_1 \sim CAT_n$  に沿って分類する。通常のツイートや該当するカテゴリがない場合は分類不可という新たなカテゴリ  $CAT_{n+1}$  として分類する。そして、Lowest ユーザのツイートも同様に分類する。

#### 4.4.2 カテゴリ特徴ベクトル

それぞれのツイートをカテゴリ化した結果から以下のような  $n+1$  次元ベクトルを得ることができる。

$$\mathbf{C}_{user} = (C_1, C_2, \dots, C_n, C_{n+1})$$

$C_i$  は  $CAT_i$  のカテゴリに分類されたリツイート数である。このベクトルのことをカテゴリ特徴ベクトルと呼ぶ。それぞれのユーザのカテゴリ特徴ベクトルを用いて、センターユーザとのコサイン類似度を求める。2 ユーザ間のコサイン類似度は以下の式で導出される。

$$\frac{\mathbf{C}_{userA} \cdot \mathbf{C}_{userB}}{|\mathbf{C}_{userA}| \times |\mathbf{C}_{userB}|}$$

この値が高いほどユーザ間の興味は似ていると考えることができる。

表 4.1: 評価結果

User	Number of retweet	Number of good retweet	Ratio of good retweet	Cosine similarity
<i>A</i>	11	8	0.7273	0.0154
<i>B</i>	11	10	0.9090	0.0123
<i>C</i>	26	20	0.7692	0.0175
<i>V</i>	18	6	0.3333	0.0092
<i>W</i>	5	1	0.2000	0.0003
<i>X</i>	2	1	0.5000	0.0542
<i>Y</i>	19	0	0	0
<i>Z</i>	1	1	1	0.0386

### 4.4.3 結果

#### コサイン類似度

Highest ユーザ, *A*, *B*, *C* 及び Lowest ユーザ, *V*, *W*, *X*, *Y*, *Z* についての検証結果が表 4.1 である. 表 4.1 から, Highest ユーザのコサイン類似度は Lowest ユーザよりも高い結果となっていることがわかる. 例外的にユーザ *X*, *Z* は値が高くなっているが, 彼らのリツイート回数は少ないため, 適切なユーザではない. このように, オーバーラップグラフから見つかったユーザは確かにセンターユーザと同じ興味を持っているユーザであることがわかる.

#### ハブユーザ

ユーザ *V*, *Y* はリツイート回数は多いが, センターユーザの興味ある内容についてはリツイート数が少ない. 彼らはいろいろな話題についてリツイートをするハブユーザとなっており, ハブユーザとしては有能である. だが, 先行研究ではセンターユーザの興味ある内容をリツイートするユーザを探しているため, 不適當である.

## 第5章 提案システム

### 5.1 概要

Twitter において、フォロワーの数が多くなるとタイムラインに表示されるツイート数は膨大になる。そのため、自分に有益な情報が雑多な情報に埋もれてしまうことがある。リツイートされたツイートはリツイートユーザによって有益であると判断されたものであることを考慮すると、リツイートを用いることで与えられるツイート群から有益なものを抽出できると考えられる。そこで、本システムは対象ユーザのタイムラインをリアルタイムに監視することで、そのユーザにとって有益なリツイートを提供するユーザを推薦する。基本的な考え方は4章と同様で、リツイートを共有しているユーザ達は同一の興味を持っていると推測する。そのため、自分とより興味が似ている推薦ユーザから与えられるリツイートは、自分にとっても興味ある内容であると考えられる。ユーザがリツイートをするたびに、そのツイートをリツイートしたユーザリストを取得し、サーバ内に保存することで推薦ユーザを更新する。本システムは、推薦ユーザから与えられるツイート及びリツイートを2つの手法でユーザに提示する。

### 5.2 情報提示方法

#### 5.2.1 リスト管理システム

本システムでは Twitter から提供されるリスト機能を用いてユーザに情報を提示する。リスト機能は、リストに登録したユーザのツイートのみを表示できる機能である。Twitter の Web クライアントでは登録ユーザの行ったリツイートは表示されないが、他の Twitter クライアントによっては見ることができる。そのため、ある特定のユーザ達のみのツイートを見たい時にリストで管理する方法は有能である。また、リストに追加するユーザをフォローしている必要はない。推薦ユーザリストは非公開リストとして管理されており、ユーザがリツイートを行って推薦結果が変化したときに、非公開リストも自動で変更される。Twitter が提供している機能であるため、ユーザは Twitter にアクセスできる環境ならばいつでも推薦ユーザのツイートを見ることができる。

#### 5.2.2 リツイート専用 Viewer

推薦ユーザのリツイートのみを重点的に監視するために、ユーザインターフェースとして図 5.1 のようなリツイート専用 Viewer を作成した。この Viewer にはユーザが行ったリツイート、推薦ユーザが行ったリツイートが表示される。また、推薦ユーザが行ったリツイートから、ユーザのタイムラインには無い興味あるリツイートを推薦し、提示している。リツイートのみを抽出して表示している Viewer であるため、Twitter におけるリスト機能よりも有益な情報のみに着目して観察することが可能となっている。



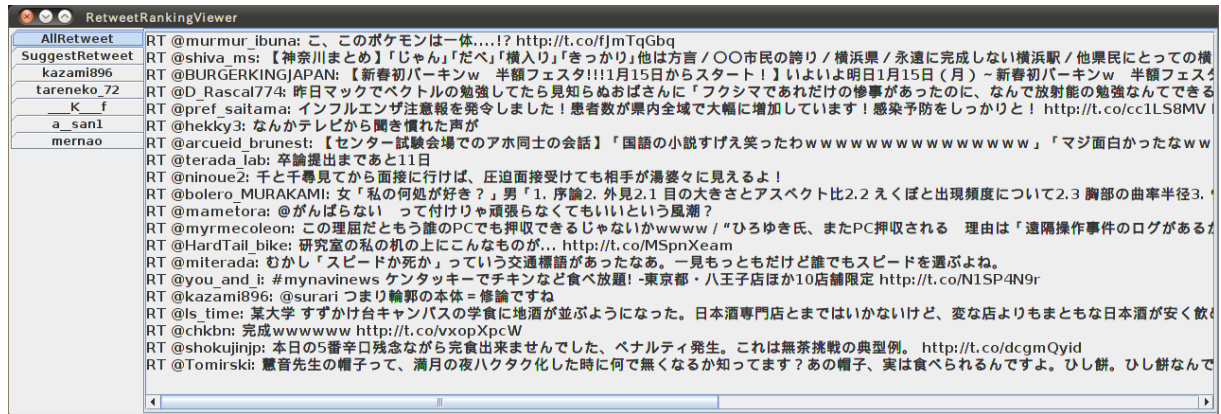


図 5.1: リツイート専用 Viewer

## 5.3 特徴

本システムを利用する場合、様々な恩恵を得ることができる。

### 5.3.1 自動リスト管理機構

ユーザが自分のタイムラインの中で見たいツイートを分けたい場合、大抵がクラスタ別にリストとしてユーザを管理する方法を取る。興味あるユーザリストを作成しようとした場合、明確に分類するユーザが決まっている場合はリスト作成は容易である。だが、分類基準が曖昧なユーザをリスト管理するのは非常に難しい。本システムではリストを自動管理しているため、ユーザの判断の曖昧さを解消している。また、リツイートを行ったら逐次推薦ユーザを更新しているため、特にユーザの手を煩わすことなく、無意識的にリストを最新の推薦されたユーザリストとして管理することが可能となっている。

リストに追加した時は興味ある内容をツイートするユーザであったが、後に実はそのユーザはあまり興味ある内容をツイートしないことが発覚する場合もある。本システムでは、蓄積されたデータを利用して推薦を行っているため、一時的な興味に左右されることがなく、興味類似度が高いユーザを推薦することができる。

### 5.3.2 未知ユーザの発見

本システムはリツイート共有情報を利用しているため、自分とは関係がない未知のユーザが推薦されることがある。このようなユーザは先行研究においても発見することができた、潜在的に自分と興味が似ているユーザである。通常、自分の興味ある内容をツイートまたはリツイートするユーザをフォロワー以外から探そうとする場合は難しい。本システムを利用することでそのようなユーザを発見し、フォローの補助をすることができる。

## 第6章 実装

### 6.1 開発環境

リツイートユーザ情報取得に TwitterAPI を使用した。具体的に利用したリソースは以下のものである。

表 6.1: 使用した API

リソース	取得データ
GET statuses/retweeted_by_user	特定のユーザの行ったリツイート
GET statuses/retweeted_by	リツイートを行ったユーザリスト

なお、本システムは APIversion1.0 を使用している。本論文執筆時現在、TwitterAPI は version1.1 に移行期間である。そのため、使用している API の中には今後利用できなくなるリソースがあるため、適宜利用できるように手法を変えなければならない場合がある。また、API を使う際に JAVA ラップである Twitter4j<sup>4</sup> を利用した。

### 6.2 システム運用

運用中の処理の流れは図 6.1 である。システム運用中は常にユーザのタイムラインを監視している。ユーザがリツイートを行ったとき、Twitter4j を通して Twitter からリツイートユーザリストを取得する。取得してきたリツイートユーザリストを次節で述べる推薦ユーザ導出アルゴリズムに使用し、推薦ユーザを更新する。もし、推薦ユーザが変化した場合、Twitter 上でのリストを変更し、使用ユーザにリプライの形で変更した旨を伝える。また、リツイート Viewer の推薦ユーザリストも変更する。その際に、推薦ユーザの過去に行ったリツイートを取得する。

### 6.3 推薦ユーザ導出アルゴリズム

サーバ上にはユーザのリツイート共有回数リストが保存されている。ここで、共有回数をスコア  $s$  とする。スコアは以下のように、ユーザ名  $u$  とセットでリストとして保存されている。

$$\mathit{ScoreList} = ([u_1, s_1], [u_2, s_2], \dots, [u_n, s_n])$$

リツイートを共有している回数が多いユーザは、より興味が似ているユーザであり、推薦ユーザと考えることができる。ユーザがリツイートを行った場合、推薦ユーザは以下の手順で更新される。

1. リツイートユーザリスト取得

<sup>4</sup><http://twitter4j.org/ja/index.html>

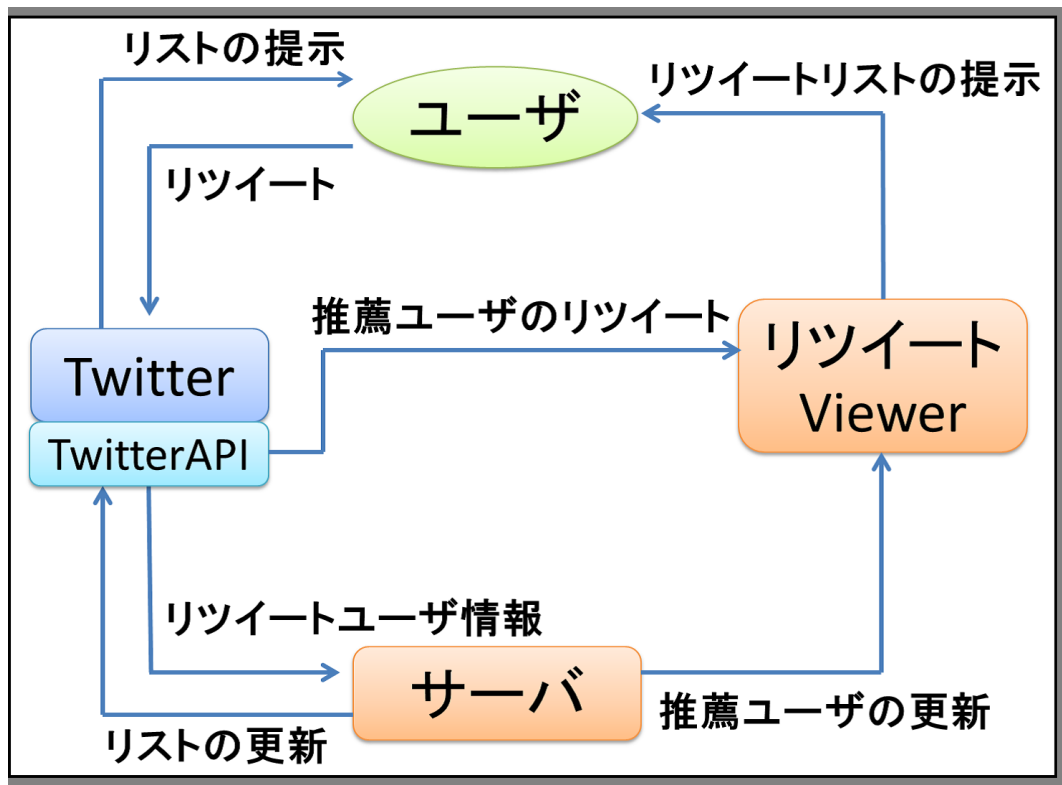


図 6.1: システム図

2. 共有回数データを更新
3. 共有回数上位5人を推薦ユーザ

まず、あるツイートをリツイートをした  $k$  人のユーザのリスト

$$User = (u_1, u_2, \dots, u_k)$$

を Twitter から取得する。次に、サーバ上に保存しているユーザのリツイート共有回数リストを更新する。 $User$  のそれぞれ該当するスコア  $s$  を取得し、加算する。もし、リツイートユーザ  $u_i$  が  $ScoreList$  に存在していない場合は、新たに  $[u_i, 1]$  を  $ScoreList$  に追加する。そして、反映されたスコアの上位5ユーザを新たな推薦ユーザとして推薦する。

## 6.4 初期状態生成

本システムを運用する下準備として、ユーザは本システムのアプリ認証を行う。この認証には、OAuth 認証 (6.5) を使用する。認証後はサーバから  $UserStream$  を使用することで Twitter と接続を張り、ユーザのタイムラインを監視する。また、初期にデータがない状態から運用を始めると、適切な推薦結果を得られるまで時間がかかってしまう。そのため、運用開始時に利用ユーザの過去20件分のリツイートを元に仮の推薦ユーザを導出しておく。そして、仮推薦ユーザを Twitter 上のリストに追加、及びリツイート Viewer に反映させる。

## 6.5 OAuth 認証

本システムを使用するにあたって、ユーザ認証として OAuth 認証を利用している。OAuth 認証では直接 Twitter でのパスワードを利用せず、与えられるアクセストークンを利用するため、パスワードの保護に繋がる。例として、Twitter のアプリケーションを OAuth 認証を行って利用する手順を以下に示す。

1. アプリを Twitter に登録して、コンシューマーキーとコンシューマーシークレット取得
2. ユーザがアプリを起動
3. アプリが Twitter にコンシューマーキー、コンシューマーシークレットを渡す
4. Twitter がリクエストトークン、リクエストシークレットを発行
5. アプリはユーザにリクエストトークン、リクエストシークレットを渡し、Twitter での認証ページに誘導
6. ユーザが認証後、認証鍵を取得でき、ユーザはアプリにリダイレクトされる
7. 認証鍵とリクエストトークン、リクエストシークレットを Twitter に渡す
8. Twitter からアクセストークンとアクセスシークレットを取得

今回は、Twitter developers<sup>5</sup>にてアプリを登録し、アクセストークン、アクセスシークレットを得た。

---

<sup>5</sup><http://dev.twitter.com/>

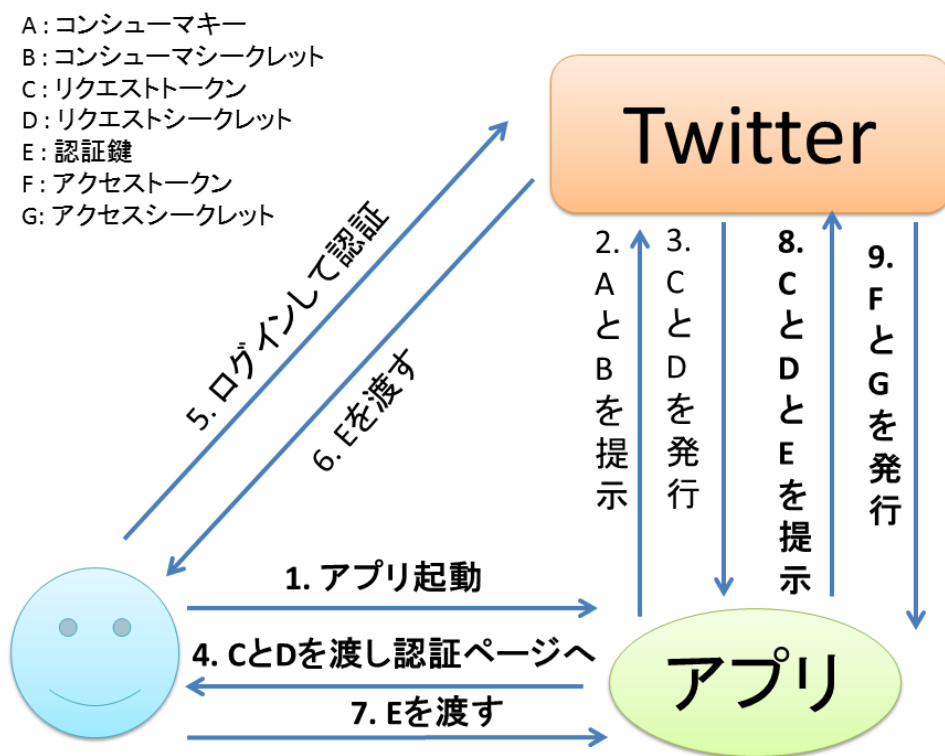


図 6.2: OAuth 認証例

## 第7章 評価実験

### 7.1 概要

リツイートを共有しているユーザ達が同一の興味を持っていることは、先行研究にて示している。だが、提案システムを使用することで、本当に興味ある内容を得られやすくなったか評価する必要がある。

### 7.2 手法

本研究の評価実験として、推薦ユーザから与えられるリツイートは通常のタイムラインに比べてユーザの興味ある内容をどの程度含んでいるか評価した。手法としては、以下の通りである。

1. システムの登録
2. システム運用 (1 か月)
3. ツイートの評価
4. 有益率の導出

まず、本検証システムを利用できるようにするために、被験者には Twitter にアプリの登録をしてもらった。次に、1 か月間の間通常通り Twitter を利用してもらった。被験者は Twitter のリストのページから、いつでも推薦されたユーザ及びツイートを確認できる状態であった。

1 か月の運用後、被験者にツイートの評価させた。評価に使用したツイートは、被験者のタイムライン上にあるツイート及びリツイート、最終的に推薦されたユーザから与えられるリツイート、推薦対象外となったユーザのリツイートである。ここで言う、推薦対象外ユーザとは、リツイート共有回数リストには入っているが、システムの最終的な推薦結果には選ばれなかったユーザのことである。それぞれのツイートを被験者の主観で興味ある、面白いと思うツイートは True, そうでないツイートは False という 2 択で評価させた。評価時には、全てのツイート、リツイートを混ぜた状態、且つツイート内容のみ提示することで、リツイートかどうかを伏せた状態で評価させた。評価結果から、それぞれにどれだけ興味ある内容が含まれているかという有益率を導出した。

### 7.3 データセット

被験者は 本論文著者を含む本学学生 5 名である。いずれのユーザもリツイートを日常的に行っているユーザである。また、ツイートの評価は以下のものからランダムに 100 件抽出したものを利用した。

- 被験者のタイムラインから取得したリツイートを含むツイート 800 件
- 推薦されたユーザ達の行ったリツイート
- 推薦対象外のユーザ達の行ったリツイート

## 7.4 結果

表 7.1: 評価結果

ユーザ	タイムライン有益率	推薦ユーザリツイート有益率	対象外ユーザリツイート有益率
A	0.14	0.48	0.34
B	0.10	0.26	0.20
C	0.11	0.17	0.14
D	0.05	0.22	0.29
E	0.28	0.24	0.39

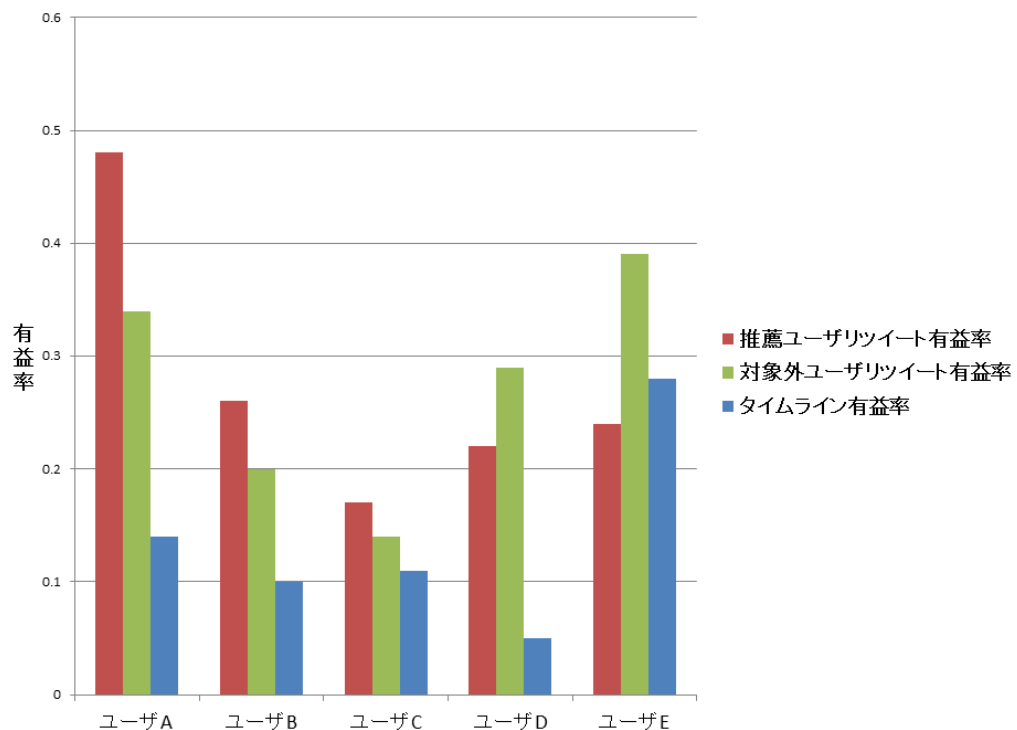


図 7.1: 有益率結果

表 7.1, 図 7.1 が通常のタイムライン, 推薦ユーザから与えられるリツイート, 推薦対象外ユーザから与えられるリツイートの有益率を示したものである. タイムライン有益率は以下の式 7.4 で求められる.

$$\text{有益率} = \frac{\text{良ツイート} + \text{良リツイート数}}{\text{ツイート数} + \text{リツイート数}}$$

また, 推薦ユーザ及び対象外ユーザリツイート有益率は, 式 7.4 において, ツイート数及び良ツイート数を 0 とし, リツイートのみを考慮した式で求められる.

それぞれの有益率は, 取得した 100 件中何件のツイート及びリツイートが興味ある内容であることを示している. そのため, 有益率が高いほど興味ある内容が含まれていることになっている. また, 表 7.2 は取得した

表 7.2: タイムラインにおける評価結果

ユーザ	ツイート数	良ツイート数	ツイート有益率	リツイート数	良リツイート数	リツイート有益率
A	88	9	0.10	12	5	0.42
B	84	7	0.08	16	3	0.19
C	88	10	0.11	12	1	0.08
D	76	3	0.04	24	2	0.08
E	81	19	0.23	19	9	0.47

タイムライン情報をツイート、リツイート別に分けて有益率を示しているものである。式 7.4 において、それぞれツイート、リツイートのみを考慮することで導出される。通常のツイートはすべて不必要である情報とは限らず、興味ある内容も含まれる。そのため、タイムライン中のツイートの有益率もタイムライン全体の有益率を見る際に重要になる。

ユーザ A, B, C の結果を見ると、推薦ユーザリツイート有益率が最も高く、次に対象外ユーザリツイート有益率、そしてタイムラインの有益率が一番低くなっていることがわかる。つまり、システムで推薦されたユーザのリツイート群を見ているほうが、タイムラインを見ているよりも興味ある内容に出会いやすいということである。タイムラインと推薦ユーザから与えられる情報の両方を見る場合でも、事実上運用中の情報有益率全体は向上する。

ユーザ D の場合、推薦対象外ユーザリツイート有益率が最も高い結果となった。推薦対象外ユーザは、リツイート共有回数が 1 回以上ではあるが、推薦ユーザよりは共有回数が少ないユーザである。ユーザ D に関しては、ユーザの持つ興味範囲が広く、推薦対象外ユーザのリツイート内容についてもカバーしている可能性がある。その場合、興味を絞った推薦ユーザよりも、少しずつ興味範囲が分散している対象外ユーザのほうが有益率は高くなると考えられる。

ユーザ E の場合は対象外ユーザリツイート有益率が最も高く、推薦ユーザリツイート有益率が最も低い結果となった。表 7.2 をみると、もともとタイムライン中のツイート有益率は他のユーザよりも高くなっていることがわかる。つまり、ユーザ E のタイムラインには、リツイートはしないがユーザ E にとって興味あるツイートを行うユーザが多いと考えられる。また、推薦されたユーザはユーザ E の興味ある内容以上に別のジャンルについてリツイートを行っているかと推察できる。

以上のことから、通常のタイムラインより、本システムが推薦したユーザのリツイート情報のほうがユーザにとって有益な情報を含みやすいといえる。



## 第8章 重みづけによる推薦ユーザ変化シミュレーション

### 8.1 概要

現在, 提案システムでは推薦ユーザを決定する際にリツイート共有回数の多さを比較している. この手法の場合, 過去のリツイート共有回数を引き継いでいるため, 以前は推薦ユーザとして適切であったが, 現在はそうでなくなったユーザの対処が難しい. そこで, 過去のデータに重みづけを行うことで, 新規の情報をより反映した推薦ユーザを導出することができる. 今回は推薦ユーザを更新する際にある一定の重み定数を過去のデータに掛けることで, 推薦結果ユーザがどのように変化するかシミュレーションを行った.

### 8.2 データセット

本研究の検証実験におけるユーザ B は検証実験ユーザの中でも, システム運用中最もリツイート回数が多かったため, 今回のシミュレーション対象ユーザとした. 提案システム運用中に保存してある, 対象ユーザのリツイートデータ (リツイートをしたユーザリスト) を使用した. また, 重み定数  $\mu$  を 0.95 から 0.01 刻みで 0.99 までとしてシミュレーションを行った.

### 8.3 手法

本シミュレーションでは, 以下の式 8.1 のように, 推薦ユーザを更新する際に, 過去のリツイート共有回数データに全体に重み定数  $\mu$  を掛ける.

$$Score = Score * \mu \quad (8.1)$$

その後, リツイートしたユーザの共有回数を 1 加算する. また, 初めて登場するリツイートユーザの Score は 1 とする. システム運用中に保存した, それぞれのリツイートに対してリツイートをしたユーザリストを使用し, 重みをかけた場合の推薦ユーザ及びそのスコアの変化をシミュレーションした.

### 8.4 結果

表 8.1 は最終的に得られた推薦ユーザリストとそのスコアを示している. 重み無しの場合, このスコアはリツイート共有回数と同値である. このスコアが低いほど, 新規ユーザのデータが生かされやすいこととなっている.

この中で注目すべきところは, 以下のような条件を満たすスコアである.

$$Score < 1.00/\mu \quad (8.2)$$

表 8.1: シミュレーション結果 (推薦ユーザリスト, スコア)

スコア順位	$\mu = 1.00$ (重みなし)	$\mu = 0.99$	$\mu = 0.98$	$\mu = 0.97$	$\mu = 0.96$	$\mu = 0.95$
1位	<b>F</b> (19)	<b>F</b> (5.66)	<b>F</b> (2.91)	<b>F</b> (1.83)	Y (1.73)	Y (1.67)
2位	<b>G</b> (12)	<b>I</b> (2.70)	Y (1.88)	Y (1.80)	<b>F</b> (1.26)	X (1.10)
3位	<b>H</b> (8)	Z (2.26)	<b>I</b> (1.74)	X (1.39)	X (1.23)	Z (1.04)
4位	<b>I</b> (7)	<b>G</b> (2.25)	Z (1.65)	Z (1.34)	Z (1.16)	W (1.01)
5位	<b>J</b> (6)	<b>H</b> (2.23)	X (1.56)	<b>I</b> (1.25)	W (1.03)	V (1.00)

これは、スコア結果に  $\mu$  を掛けた値が 1.00 未満になるスコアである。スコア計算には前述した式 8.1 を使用する。推薦ユーザのスコアがこの条件を満たす場合、次にユーザがリツイートをしたとき、初めて出現したリツイートユーザも推薦対象となりうる。この場合、推薦ユーザがリツイートを多く共有しているユーザであるという定義から外れてしまい、適切ではない。そのため、重みを掛ける場合、推薦ユーザとなりうるスコアの閾値が低くなりすぎないように重みを考えるべきである。

次に、推薦ユーザリストに着目してみる。太字で表示されているユーザ F, G, H, I, J は重みを掛けなかった場合の推薦ユーザである。 $\mu = 0.98$  以下の結果を見ると、最大でも 2 ユーザのみが残る結果となった。特に  $\mu = 0.95$  では、推薦ユーザはすべて別のユーザとなった。推薦結果が全て別のユーザになるということは、本来残っていてほしいユーザまでも排除してしまっている可能性がある。

このシミュレーション結果からは、スコアに掛ける重みがどの程度推薦ユーザの変化に影響するかを理解することができた。また、推薦ユーザスコアの閾値など、共通して則っていなければならない条件は複数存在した。具体的にどの重みが良いかは、利用ユーザの考え方によって変わってくる。そのため、利用ユーザの求める重みの掛け方をどのように決定していくかを今後検討する必要がある。

## 第9章 結論

### 9.1 結論

本研究では、Twitterにおけるリツイート機能に着目して、自分に興味ある情報を与えるユーザを推薦した。ユーザが通常通りリツイートを行うだけで、自動的に推薦ユーザを更新することができる。ユーザは本研究が提供するリツイート Viewer や Twitter 上のリストを観察することで、推薦ユーザから与えられる有益な情報を見ることができる。

評価実験の結果、確かに推薦ユーザが与える情報にはユーザの興味ある内容を含みやすいことが分かった。そのため、本システムが推薦したユーザからの情報を見ることで、ユーザが得る興味あるツイート率はよくなった。

### 9.2 今後の課題

#### 9.2.1 リアルタイム性のあるユーザの興味に対する対処

タイムラインにおけるトレンドは、その時に起こっているイベントや話題によって変化が激しい。そのため、ユーザはある時には興味ある内容であったが、数日後には興味が全くなくなっているということがある。本システムのように常に蓄積されたデータを利用している場合、ユーザの潜在的な興味に対しては対応できるが、トレンドによってユーザの興味の移り変わるような瞬間的な興味には対応が難しい。この2種類の興味は相反するものであるが、ユーザにとっては両方とも重要である。前章で述べた重みづけがこの対処について有効であると考えられる。今回のシミュレーションでは、重み定数を一定とした。だが、タイムライン中のトレンドを反映させるとすると、ユーザがリツイートした時刻を考慮して、重み定数を動的に変化するとより適切な推薦ができる可能性がある。また、重みを掛けるタイミングも考慮する必要があるだろう。

## 謝辞

本研究は、電気通信大学情報理工学研究科情報・通信工学専攻コンピュータサイエンスコース寺田研究室において、寺田実准教授のご指導の下に行われました。また、本研究は電気通信大学創立 80 周年記念学術交流基金の助成を受けて行われました。国際学会参加の助成に感謝します。

寺田実准教授には、研究全般のアイデア出しや指針の検討、国際学会への論文の出し方など様々な部分でご指導をいただきました。心からお礼を申し上げます。

明星大学情報学部の丸山一貴准教授には、研究についての助言や国際学会での発表の仕方など多くのことに関してご指導をいただきました。深く感謝いたします。

寺田研究室の皆様からは、研究内容についての意見を頂いたり、作成したプログラムを利用して頂いたり感謝しております。

また、本研究の評価実験を快く引き受けてくださった小林達也君、鈴木基玄君、佐々木佳祐君、贅田将史君には非常に感謝しております。

## 参考文献

- [1] 西田 恭介, 坂野 遼平, 藤村 孝, 星出 高秀. “データ圧縮による Twitter のツイート話題分類”. DEIM Forum 2011 A1-6.
- [2] danah boyd, S. Gokler, and G. Lotan. “Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter ”. Proceedings of the 43rd Hawaii International Conference on System Sciences, 2010.
- [3] Sofus A. Macskassy and Matthew Michelson. “Why Do People Retweet? Anti-Homophily Wins the Day!”. Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), 2011.
- [4] Bongwon Suh, Lichan Hong, Peter Pirolli and Ed H. Chi. “Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network ”. Second IEEE International Conference on Social Computing (SocialCom), pp. 177–184, 2010.
- [5] John Hannon, Mike Bennett, and Barry Smyth. “Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches”. Proceedings of the fourth ACM conference on Recommender systems (RecSys ’10), pp. 199–206, 2010.
- [6] Matthew Michelson, and Sofus A. Nacskassy. “Discovering users’ topics of interest on twitter: a first look”. AND’10 Proceedings of the fourth workshop on Analytics for noisy unstructured text data, pp. 73–80, 2010.
- [7] Qing Chen, Timothy Shipper, and Latifur Khan. “Tweets Mining Using WIKIPEDIA and Impurity Cluster Measurement”. Proceedings of the Intelligence and Security Informatics (ISI), pp. 141–143, 2010.
- [8] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. “Analyzing User Modeling on Twitter for Personalized News Recommendations”. Proceedings of the 19th international conference on User modeling, adaption, and personalization, pp. 1–12, 2011.
- [9] Minoru Yoshida, Shin Matsushima, Shingo Ono, Issei Sato, and Hiroshi Nakagawa. “ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management”. CLEF 2010 Labs WePS, 2010.
- [10] Cristian K. dos Santos, Alexandre G. Evsukoff, Beatriz S. L. P. de Lima, and Nelson F. F. Ebecken. “Potential Collaboration Discovery using Document Clustering and Community Structure Detection”. Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management (CNIKM’09), pp. 39–46, 2009.

- [11] David Auber, Yves Chiricota, Fabien Jourdan, and Guy Melançon. “Multiscale Visualization of Small World Networks”. INFOVIS’03 Proceedings of the Ninth annual IEEE conference on Information visualization, pp. 75–81, 2003.
- [12] Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang, and Zhong Su. “Understanding Retweeting Behaviors in Social Networks”. CIKM’10 Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 1633–1636, 2010.
- [13] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. “Measuring User Influence in Twitter: The Million Follower Fallacy”. Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM), 2010.
- [14] Yusuke Ota, Kazutaka Maruyama, and Minoru Terada. “Discovery of interesting users in Twitter by overlapping propagation paths of retweets”. Proceedings of the 2012 IEEE/WIC/ACM international Conference on Web Intelligence (WI), pp. 274–279, 2012.