平成 23 年度

筑波大学情報学群情報科学類

卒業研究論文

題目
マイクロブログにおけるユーザ間でのトピックの広がり方の可視化手法

主専攻　　情報システム主専攻

著者　　武田慎之介

指導教員　三末和男　志築文太郎　高橋伸　田中二郎

要　旨

近年爆発的にユーザ数を増やしているマイクロブログは、コミュニケーションに用いるために利用しているというユーザも多い。代表的なマイクロブログにおいても、日々様々なトピックが発生して広がっている。本研究では、あるユーザとそのユーザがフォローしているユーザによるコミュニティにおいて、トピックがどのように広がるかを直感的に観察できるようにするための手法を提案する。本研究では、ツイートを収集してそこからトピックを抽出、更にはユーザのトピックへの貢献度の時間変化を考えて、最終的にユーザがトピックの広がる様子を直感的に観察出来るようにするための可視化までのプロセスを含んだ、可視化手法を提案する。実際にこの手法を用いた実装を行い、筑波大生と推測されるユーザによるコミュニティにおいて、トピックが広がる様子を観察した。

目 次

第 1 章　序論 **1**
1.1　ソーシャル・ネットワーキング・サービスの現状 1
1.2　本研究の目的 2

第 2 章　関連研究 **3**
2.1　マイクロブログにおけるトピックに関する研究 3
2.2　トピックの抽出に関する研究 4

第 3 章　課題の分析 **5**
3.1　現実世界におけるトピック 5
3.2　マイクロブログにおけるトピック 5
3.3　トピックの広がり 6

第 4 章　提案手法 **7**
4.1　全体の流れ 7
4.2　データの収集 7
4.3　データの加工 8
4.3.1　トピックの決定 8
4.3.2　各単位時間のユーザ貢献度の決定 9
4.4　広がり方の可視化 9
4.4.1　ユーザの表現 9
4.4.2　各ユーザの貢献度の表現 9
4.4.3　アニメーションによる広がる様子の表現 9
4.4.4　ユーザ描画位置決定 10
4.4.5　色による区別 10

第 5 章　実装 **12**
5.1　環境 12
5.1.1　サーバ・クライアントモデルの利用 12
5.1.2　サーバ環境 12
5.1.3　クライアント環境 12
5.2　データの収集 13
5.2.1　収集の対象ユーザ 13

5.2.2 ツイートの取得 13
5.3 データの加工 14
5.3.1 トピックの抽出 14
名詞によるトピックの決定 14
ハッシュタグによるトピックとそれに対するユーザの貢献度の決定 . 15
5.3.2 ユーザ描画位置の決定 16
多次元尺度構成法 16
5.4 描画 16
5.4.1 HTML5/Javascript 16
5.4.2 データの取得と描画 17

第 6 章 ユースケース **18**
6.1 実行例 18
6.1.1 ランダムな配置と多次元尺構成法による配置 18
6.1.2 アニメーションの観察 18

第 7 章 結論 **24**

謝辞 **25**

参考文献 **26**

図 目 次

1.1 一般的な Twitter のタイムライン 1

4.1 ハッシュタグ (#iplab) 付きのツイートの例 8
4.2 ユーザの表現と貢献度の表現 10
4.3 広がる様子の表現 11

6.1 ランダムに配置した例 19
6.2 多次元尺度構成法を用いて配置した例 20
6.3 多次元尺度構成法を用いて配置した例の拡大 21
6.4 トピックの広がる様子 1 (図 6.2 をアニメーションしたものの抜粋) 22
6.5 トピックの広がる様子 2 (図 6.2 をアニメーションしたものの抜粋) 23
6.6 トピックの広がる様子 3 (図 6.2 をアニメーションしたものの抜粋) 23

第1章　序論

1.1　ソーシャル・ネットワーキング・サービスの現状

近年、世界的に Twitter[1](図 1.1) や Facebook[2]を始めとしたソーシャルネットワーキングサービス (SNS) を利用するユーザが大幅に増え、特に Twitter においては、全世界でのアクティブユーザが 2011 年の 9 月の時点で 1 億人を超え[3]、更に 2011 年 12 月 9 日には、全世界で 1 秒あたり 25,088 ツイートを記録するまでになった[4]。

図 1.1: 一般的な Twitter のタイムライン

Twitter は、近年特にユーザ数を爆発的に増やしている。なぜ多くの人に Twitter は受け入れられるのか。あるアンケート結果[5]によると、Twitter を利用する理由は「ゆるいつながりをも

[1]http://twitter.com/
[2]http://www.facebook.com/
[3]http://blog.jp.twitter.com/2011/09/1.html
[4]https://twitter.com/twittercomms/statuses/146751974904311808
[5]http://research.netmile.co.jp/voluntary/2010/pdf/201004_2.pdf

つ」ために役に立つからだとある。Twitter を利用するユーザは、人とのつながりを意識しているユーザが多いということが分かる。

これは、すなわち多くのユーザが、マイクロブログにおいて自分自身と自分がフォローしているユーザによるコミュニティでの、ゆるいつながりが役に立つから、これらを利用していると考えられる。

1.2 本研究の目的

本研究の目的は、マイクロブログにおけるユーザ間で、トピックがどのように広がるかを観察することである。

現実社会においては、例えば自分とその周りの友人らで形成されるコミュニティを分析しようとした時、そもそも誰がそのコミュニティに属するかが不明瞭であったり、またコミュニティ内での会話を全て記録することも困難であるので、現実社会におけるコミュニティを分析することは、一般に非常に困難であると知られる。

代表的なマイクロブログである Twitter では、実際のつながりがある友人同士がフォロー関係にある場合が多い。ここで、Twitter において自分自身と自分がフォローしているユーザで形成されるコミュニティを考えると、これは現実社会での自分と自分の友人などからなるコミュニティに非常に近いコミュニティである場合が多い。

すなわち、Twitter に代表されるマイクロブログにおいて、自分自身と自分がフォローしているユーザによって形成されるコミュニティを分析することは、現実社会での自分自身と自分の周囲の友人らによるコミュニティを分析することにも繋がる。

コミュニティをトピックという観点から分析することに繋がるように、本研究ではトピックが広がる様子を直感的に観察するための可視化手法を提案する。また、本研究では、実際に提案する可視化手法を用いた実装例も作成した。

第2章　関連研究

2.1　マイクロブログにおけるトピックに関する研究

Java らは、Twitter を観察・分析し、人々は日々の出来事について会話したり、情報を自分のフォロワー達と共有するために Twitter を利用するということを発見した [5]。

Naaman らは、実際にこのことを量的に検証し、Twitter のソーシャルネットワークにおける活動をより深く理解するために、350 ユーザのツイートを、手作業でカテゴリごとに分類して分析を行った [6]。これによると、Twitter においては、ツイートの 20%強程度が情報共有、25%弱程度が意見や主張、25%弱程度が思いつきの考え (ランダムな考え)、そして 40%程度が自分自身の今をに関するものであった。

Honeycutt らもまた、Twitter においてユーザ間での会話によるコミュニケーションが多くなってきていることを指摘している [7]。また Honeycutt らは、特に、@を用いた返信の機能による会話について分析している。

Zhang らは Twitter によく似たマイクロブログサービスである Yammer において、Fortune 500 に載っている企業のユーザがどのようにマイクロブログサービスを利用するかを調べた [8]。ユーザは Yammer においてどのような方法で重要な情報に注目するかという調査では、実に 60%以上のユーザがフォローする人を選ぶことによって、重要な情報に注目しやすいように調整すると回答した。そして、そのフォローする人をどのように選ぶかという調査に対しては、多くのユーザはフォロー機能について、誰でも闇雲にフォローすることは無く過去に自分が興味があるトピックについてポストしたユーザを最も多くフォローする、という調査結果を得た。

Zhang らの研究 [8] では、フォローする人を増やしすぎるとノイズに相当する情報も多くなるということが言及されていたが、Ehrlich らもまた、マイクロブログがリアルタイムな情報源として用いられてきている一方で、同時に入ってくる情報の量も飛躍的に増加していることを指摘している [9]。

Bernstein らは、マイクロブログにおけるトピックを分析するための研究として、Twitter のトピックブラウジングのための Eddi という システムを開発した [4]。これは、トピックを Twitter 上で呟かれている特徴語により決定し、その特徴語を含むツイートをトピックに属するツイートしてツイートを分類し、一般的なタイムラインのようにまとめて表示するものである。

この研究は、無作為に全ツイートから 5%程度を取得したツイートにおいてトピックごとに分類して表示するもので、Twitter 全体において、トピックのトレンドを見るためのものである。

2.2 トピックの抽出に関する研究

マイクロブログにおいてトピックを認識・抽出するためには、まずはツイートを分類する必要があるが、そのための研究として、Ramage らによる LDA を用いて Twitter におけるツイートにラベル付けして分類する研究 [1] がある。

他にも、関連語によるトピックの決定については、係り受けを考慮してブログから関連語を抽出する数原らによる手法 [2] も、トピックの抽出手法として参考となる。

また、トピックの抽出について、情報検索の分野においては固有名詞はパラグラフ程度の長さのテキストのトピックを表すものとして適していることが知られている [12][13]。

しかし、候補となりうる名詞が非常に多く、トピックを表す名詞を正確に選ぶことが困難であるケースが多いが、Bendersky らは、正確さを改善するために、名詞の重要度をランク付けするために機械学習を用いた [12]。

トピックの抽出について時間軸を含めて考えると、Twitter からのリアルタイムでのトピックの抽出手法としては Cataldi らによる手法がある。

これは、トピックはマイクロブログにおけるコンテンツの一部で、コンテンツは更に発信者によって影響を受けることから、Page Rank アルゴリズム[1]を元にした発信ユーザの重み付けと、TF-IDF を組み合わせて、最近急に出現頻度が増えてかつ、発信者が Page Rank の高いユーザであれば新しいトピックが生まれたと考える。

この Cataldi らの研究は、リアルタイムでのトピック抽出を可能とした点で興味深いが、あくまでもトピックの出現のタイミングにのみ主眼がおかれ、トピックの出現以降の時間経過には着目していない。

Leskovec らは、9000 記事に出現するフレーズをクラスタリングし、ニュースサイクルの動的な変化、そして”meme”と呼ばれる文化的情報をトラッキングするアルゴリズムを開発し、実際に 2008 年のアメリカ大統領選挙期間中に、こ選挙に関する”meme”が時間経過と共に広がることを追跡した [10]。この研究は文化的情報である”meme”が時間経過と共に広がる様子を追跡するものであり、一方で本研究はトピックが時間経過とともに広がる様を見ることを目指すものであり、参考となる。

また、同様にトピックの時間変化に着目した研究として、Shamma らは大統領選挙大統領選挙期間中に議論されるトピックが時間と共にどのように変化していくかを、ツイート中に含まれるキーワードを TF-IDF 法により求められた特徴語を観察して、分析した [11]。

本研究は、以上の関連研究を横断的に組み合わせた研究で、マイクロブログにおいて、あるユーザとそのユーザがフォローするユーザによるコミュニティにおけるトピックを抽出し、さらにそのトピックがコミュニティ内を広がる様子を直感的に観察出来るようにするための可視化手法を提案するものである。

[1] http://infolab.stanford.edu/ backrub/google.html

第3章　課題の分析

3.1　現実世界におけるトピック

現実世界においては、様々なコミュニティで様々なトピックが発生している。一般的には、トピックとは話題になる事柄であったり出来事のことを指す。例えば、自分の友人同士のコミュニティを考えてみると、このコミュニティでも日々様々なトピックが話されている。それは、例えば日々の出来事のようなものであったり、何かに関する議論など様々である。実際のあるトピックに関する会話においては、一部の人がよく発話していたり、ある人の話から急にトピックが広がったりしている、ということが経験的に観察出来る。

このように、あるコミュニティにおいて、あるトピックが広がる様子を観察することを考えると、このためには先ずはトピックを成すための会話の内容を全て記録する必要があり、その上で分析をする必要があるが、現実世界においてはこれは現実的では無い。

3.2　マイクロブログにおけるトピック

近年マイクロブログのユーザが増えるにつれ、多くの人々がマイクロブログを日常的に利用するようになった。マイクロブログの特徴として、例えば Twitter においては 140 文字の制限があるなど、基本的には限られた文字数で発言を投稿するということがあげられる。この制限のために、人々はマイクロブログに実世界でのちょうど 1 回の発言程度の短さの投稿をしている。

また、多くのユーザはマイクロブログをコミュニケーションのために用いており [8]、実際に mention を用いた会話を初めとして、mention を用いずにも多人数で議論がされたりある話題や事柄について会話する様子が観察出来る。

Twitter では、実際の友人を Twitter 上でフォローし日常的に会話している様子もよく観察することが出来る。このような、あるユーザとそのフォローしている友人間では実際の現実世界における会話のように、様々なトピックが Twitter でも話されている。

すなわち、現実世界におけるあるトピックの会話のにように、Twitter においてもトピックが話されている。Twitter におけるツイートは、現実世界における発言とは異なり、データとして全て収集することが出来る。これを利用し、Twitter におけるトピックの広がりを観察出来るようにすることで、現実世界におけるトピックの広がりの一部を観察することが出来る。

3.3 トピックの広がり

ある時に、A さんが例えば「今日は線形代数の授業が休講らしい #coins 線形代数 」とツイートしたとする。この A さんは B さんと C さんをフォローしているとする。A さんのツイートに続いて、B さんが「休講だから課題出さなくていい！ #coins 線形代数 」、B さんが「線形代数の課題をそもそも知らない #coins 線形代数 」のようにツイートしたとすると、A さんから見ると、A さんのツイートが B さんと C さんに広がったように見える。本研究では、このように、あるユーザがフォローしているユーザ間でのトピックの広がる様子を観測する。

この場合には 「#coins 線形代数」のようにハッシュタグがついたツイートを、ハッシュタグによってタグ付けされたトピックとして抽出した例であるが、このようにトピックを抽出するためにはツイートを例えばハッシュタグを用いて、タグ付けする方法が必要となる。

以上のように、トピックの広がりを観察出来るようにするためには、対象コミュニティを決めてそのコミュニティでのツイートを収集し、そこからトピックを抽出する、そして、先の例のような、A さんのあるトピックに属するツイートの後に B,C さんが同じようにツイートすると、A さんからはトピックが広がったように見える、ということを実際に可視化する手法が必要となる。

第4章　提案手法

一般的なユーザは、Twitter をタイムラインによる表示 (図 1.1) で使用している。しかし、このタイムラインによる表示では、ある時点でユーザがどのトピックにどの程度参加しているかという、トピックの時間経過による変化を一目で見て理解することが困難であり、トピックの広がる様子を直感的に観察することが出来ない。

本研究では、この問題を解決するために可視化というアプローチをとり、トピックが広がる様子が直感的に観察出来る可視化手法を提案する。

本研究では、あるユーザとそのユーザがフォローしているユーザによるコミュニティにおいて、ツイートを収集してそこからトピックを抽出、更には各ユーザのトピックへの貢献度の時間変化を考えて、最終的にユーザがトピックの広がる様子を直感的に観察出来るようにするための広がる様子の可視化までのプロセスを含んだ、可視化手法を提案する。

4.1　全体の流れ

提案する手法は、全体の流れとして主に 3 つの手順による。

先ず、あるユーザとそのユーザがフォローしているユーザによって形成されるコミュニティの、ツイートのデータを収集する必要がある。続いて、そのコミュニティ内で代表的であったトピックを抽出し、各ツイートがどのトピックに属するか、また各ユーザがどの程度トピックに参加するかを導き出す。そして、これらの操作により、実際に可視化をするために必要なデータへと加工を行い、最後にこれをもとにして描画を行う。

対象とする時間の、最初のツイートがされた時刻から最後のツイートがされた時刻の間をいくつかに均等に区切り、この均等に区切られた時間を単位時間とする。この単位時間ごとに、トピックに参加するユーザの状態が変化している。すなわち、各単位時間ごとに、その単位時間の時点でのトピックに参加するユーザの状態を表現した図を、一つずつ描くことが出来る。この各単位時間ごとの図をアニメーションでなめらかに繋いだものが、最終的に描画される。

以下に各手順を詳しく述べる。

4.2　データの収集

データを収集する時点においては、どのツイートがどのトピックに属しているかといった情報が無いため、トピックに属するツイートのみを Twitter から取得するというようなことは

出来ない。そのため、先ずはコミュニティ内でのツイートを全て取得する必要がある。

本研究では、自分と自分がフォローしているユーザによって形成されるコミュニティを対象としているので、取得すべきツイートは、自分と自分がフォローしているユーザのツイートである。

また、ユーザを表現するために、あるユーザ自身とそのユーザがフォローしているユーザの情報も取得する必要がある。

4.3 データの加工

最終的にトピックがどのように広がるかということを可視化するために、先の手順により収集したデータを、可視化に必要な情報へと加工する手順が必要である。このデータの加工の手順により、どのユーザがどの程度、どのトピックに属するかという情報、そしてユーザを可視化する上で、どこに描画するのがよいかという描画位置の情報を、最終的に可視化を行うために求める。

4.3.1 トピックの決定

各ツイートをトピックへと分類するためには、何らかのタグによるタグ付けによって分類することを考える。このタグとしては、例えば、各ツイートの特徴語をタグとして、共通のタグがついたツイートをまとめてトピックとして分類する方法がある。 また他にも、Twitterではある話題についてツイートするときに、ハッシュタグ (図 4.1) というものを用いることが一般的であるので、これを基にしてトピックを抽出する方法がある。

どのツイートがどのトピックに属するかという分類がされてトピックが定まり、次に、トピックに属するツイートをツイートしたユーザはトピックに参加していると考えると、各ユーザがどのトピックに参加しているかということが定まる。

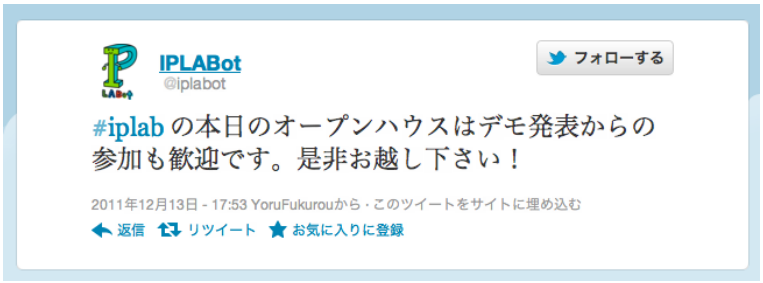

図 4.1: ハッシュタグ (#iplab) 付きのツイートの例

4.3.2　各単位時間のユーザ貢献度の決定

最終的には、各単位時間ごとに、その単位時間の時点でのトピックとトピックに参加するユーザの状態を表現した図を描くための情報が必要であるので、先の手順による、ユーザがどのトピックに参加するかに加えて、各単位時間ごとに、ユーザがどの程度トピックに参加しているかという情報を求める必要がある

この、各単位時間ごとの、ユーザのトピックへの参加の程度を、貢献度と呼ぶ。この貢献度は、単純にトピックに属するツイートをユーザが何度行ったかという指標を用いたり、または、ユーザが行ったトピックに属するツイートが何度 RT されたかという指標などを基に定める。

貢献度をどのように定めるかの指標により、各単位時間において、各ユーザのそれぞれのトピックへの貢献度、すなわちどの程度トピックへと参加しているかという情報が求まる。

4.4　広がり方の可視化

トピックの広がる様子をアニメーションで表現する。アニメーションの各フレームは、ある単位時間におけるユーザのトピックへの貢献度を表現する。以下にこのアニメーションの詳細について説明する。

4.4.1　ユーザの表現

Twitter においてはユーザアイコンを自分のオリジナルな画像にすることが一般的であり、自分自身を表すものとしてユーザアイコンを設定しているユーザも多いと考えられる。また、一般的にユーザアイコンには重複も少ない。更に、本研究においては自分と自分のフォロワーによるコミュニティを対象として、そのために普段自分がフォローしているユーザのみが最終的に描画されることからも、普段目にしているであろうユーザアイコンをユーザを表現するものとして用いる。(図 4.2)

4.4.2　各ユーザの貢献度の表現

各ユーザの、トピックへの貢献度は、ユーザアイコンの境界線の太さにより表現する (図 4.2 の「境界線」)。ユーザの貢献度が大きければ大きいほど、境界線は太く表現される。各単位時間ごとにユーザの貢献度は異なるので、各短時間ごとでユーザアイコンの境界線の太さが定まる。

4.4.3　アニメーションによる広がる様子の表現

各単位時間ごとのユーザの貢献度を表現した図をアニメーションによって繋ぐ。これにより、貢献度の時間変化がよりわかりやすく表現される。図 4.3(a) から、(b) そして (c) へとユー

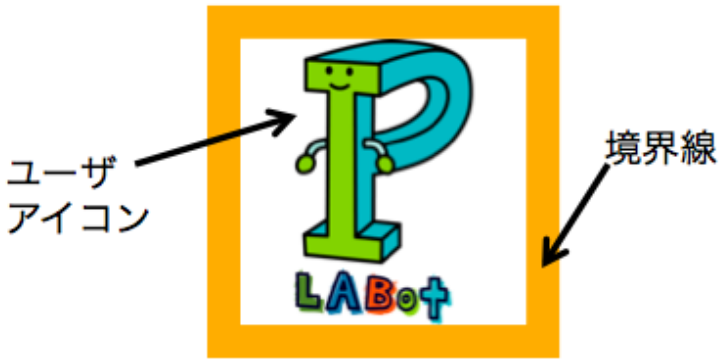

図 4.2: ユーザの表現と貢献度の表現

ザアイコンの境界線 (図 4.2 の「境界線」) が変化することを観察することで、トピックの広がる様子を観察する。

4.4.4　ユーザ描画位置決定

本研究ではあくまでもトピックの広がり方を観察するので、ユーザアイコンの位置は、最初の描画時から変化しない。ユーザアイコンを 2 次元平面にどのように配置するかについて、性質が近いユーザ同士を極力近く配置することは、近接・類同のゲシュタルト要因を利用することになり、ユーザの直感的理解を助ける。したがって、性質が近いユーザ同士を極力近く配置することが必要である。

このためには、先ずユーザの性質の近さを考える必要がある。ユーザの性質を表すものとしては、例えば誰をフォローしているかという情報、物理的な地理情報などを利用することが考えられるが、本研究においては、トピックの広がり方を可視化することで自分と自分のフォロワーによるコミュニティを分析することが目的であるので、各ユーザがどのトピックに属するかという情報をユーザの性質と考える。

この描画位置決定の方法としては、多次元尺度構成法を用いることなどが考えられる。

4.4.5　色による区別

本研究では、1 つの可視化のアニメーション中で複数のトピックの広がり方を同時に観察する。そのために、複数のトピックを直感的に区別する必要があるが、本研究では、トピックの区別に、色を用いることとした。異なる色で、異なるトピックを表現する。例えば、内容が好意的なトピックには暖色を、内容が批判的なトピックには寒色を選択するというように、色の選択をトピックの内容に準じて対応する心理的に近い色を選ぶといった工夫も考えられる。

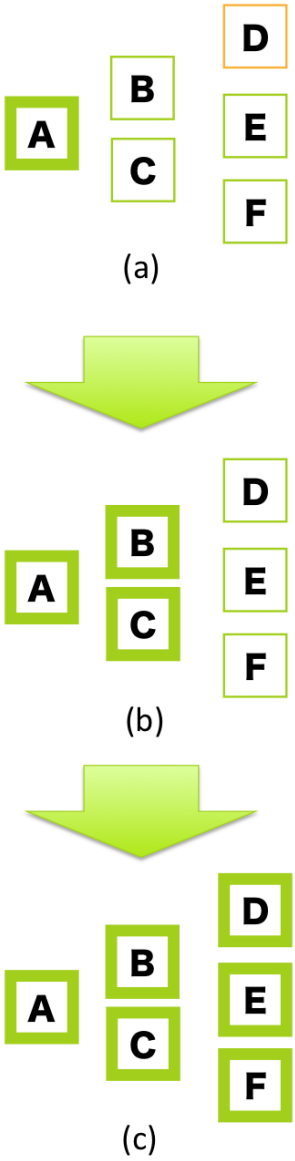

図 4.3: 広がる様子の表現

第5章　実装

5.1　環境

本研究で提案する手法を用いた実装例を、実際に開発した。本項ではその詳細を述べる。

5.1.1　サーバ・クライアントモデルの利用

本実装では、対象とするユーザと、そのフォローしているユーザのツイートを予め収集しておく。

本実装では、630 ユーザのツイートを収集するために、1 日収集するだけでも約 10,000 ツイートほどが集まる。これを継続的に収集し、さらにこのデータを加工することまで、一つの端末で行うことは現実的では無いので、ツイートの収集とデータの加工プロセスまでをサーバで行い、実際の描画の部分のみをクライアントで行うという、サーバ・クライアントモデルを採用した。

5.1.2　サーバ環境

サーバは、VMware ESXi[1]上で動く仮想マシンを用いる。その OS には Ubuntu 11.10(Linux 3.0.0-12-server x86_64 GNU/Linux)[2]を用いた。

ツイートの収集、加工プログラムは Ruby[3]を用いて作成した。プログラム自体は、ツイートの収集プログラムは Ruby1.8.7 での動作を、データを加工するプログラムは Ruby1.8.7 及び Ruby1.9.2 での動作を確認している。

5.1.3　クライアント環境

クライアントとしては、HTML5 を使うことが出来るブラウザに対応し、特に Safari5.1.1[4]での動作を確認した。

---

[1] http://www.vmware.com/jp/products/datacenter-virtualization/vsphere/esxi-and-esx/overview
[2] http://www.ubuntu.com/
[3] http://www.ruby-lang.org/
[4] http://www.apple.com/jp/safari/

5.2 データの収集

5.2.1 収集の対象ユーザ

本研究は、自分自身と自分がフォローしているユーザによって形成されるコミュニティを対象とするため、自分自身と自分がフォローしているユーザを対象としてツイートを収集する。

本実装では、ツイートを収集するための Twitter アカウント[5]を作成し、2011 年 9 月の時点で、ユーザのプロフィールから明らかに筑波大生であると推測できるユーザを探し、この作成したアカウントで合計 630 人をフォローした。収集途中、フォローしているアカウントの消滅やブロック等により、いくつかフォロワーが減ることもあったが、概ね 630 ユーザをフォローし続けるように保った。

この 630 ユーザと、フォローするために作成したアカウントが、収集の対象となるアカウントである。

本実装では、筑波大生であると推測出来るユーザをフォローしてツイートを収集する対象のユーザとしたため、注目するコミュニティは、筑波大生で Twitter を利用しているユーザということになる。

5.2.2 ツイートの取得

ツイートの取得方法としては、Twitter の REST API の user_timeline[6]から取得する方法と、Streaming API の User Streams[7]を持ちいて継続的に取得する方法の 2 つがある。

REST API は、HTTP の GET メソッドに対して一度に 20 の最新ツイートを返すものであるが、これには制限があり、他の API の利用と合わせて 1 時間最大 150 回しか利用出来ない。特に、これは最大 150 回であり、実際にはより厳しく制限がかかる。この REST API を利用する実装のほうが実装が単純で容易ではあったが、今回は収集の対象とするユーザが 630 ユーザであり、またそのユーザのほとんどが実際の筑波大生であったことからツイートの量には大きな時間的な偏りがあり、最も収集すべきツイートが多くなった夕方から深夜の時間帯においては、収集しきれないツイートがあることがわかった。

User Streams は、継続的にツイートを取得するために提供されている API である。通常の HTTP の GET メソッドではデータの転送が終わり次第サーバ側からコネクションを即座に切断するが、この User Streams での GET メソッドに対しては、Twitter のサーバはコネクションを開き続けようとする。そして、新しいツイートがツイートされた時に、その開いたままのコネクションからツイートの情報を取得することができる。そのため、コネクションが開き続けている間は、そこから取得出来るツイートの量に理論上の制限は無い。従って、User Streams による実装のほうが、取りこぼしが発生しにくいが、REST API に比べて実装は困難である。

[5] http://twitter.com/#!/ceeeeeeen
[6] https://dev.twitter.com/docs/api/1/get/statuses/user_timeline
[7] https://dev.twitter.com/docs/streaming-api/user-streams

本実装においては、REST API だと実際に取りこぼしが発生していることが確認出来たために、User Streams を用いて継続的に取得する方法を用いる必要があり、そのように実装した。

ツイートの取得においては、REST API、User Streams どちらにおいても、OAuth による認証を必要とするため、実装には Ruby OAuth GEM[8] というライブラリを利用した。

User Streams による実装で、630 ユーザのツイートを継続的に収集したが、何度かネットワーク上のトラブルなどにより、大きな切断は避けられなかった。収集の結果、何度か User Streams が途切れたために収集しきれないツイートもあったが、2011 年 10 月 20 日から 2012 年 1 月 24 日までの 73 日間で合計 73 万ツイートほどを得た。

5.3 データの加工

5.3.1 トピックの抽出

本実装では、収集したツイートに存在するトピックのうち、特に代表的なトピックの広がり方を可視化する。

名詞によるトピックの決定

名詞はパラグラフ程度の長さのテキストのトピックを表すものとして適していることが知られている [12][13]。すなわち、共通の名詞を含むツイート群は、その共通の名詞をトピックとして考えて、ツイートをトピックに分類することが出来る。

実際に、以下の処理によって名詞に基づいてトピックを決定する方法の実装を試してみた。なお、ツイートを形態素に分解して名詞を取り出すために、形態素解析エンジンの MeCab[9] を利用した。

1. 全てのツイートを形態素に分解し、名詞を抜き出す。
2. 全てのツイートの名詞の出現回数を数える。
3. 名詞を出現回数の多い順にソートする。
4. 上位の名詞を、このコミュニティにおける代表的なトピックを定義するための名詞とする。
5. この名詞を含むツイートを、この名詞によるトピックに属するものと考える。

実際にこの手法を実装するために、60,000 ツイートを形態素に分解し、名詞の出現回数を数えた。その結果から、”@”や”RT”、記号、”http”等の URL の一部を除いた結果の上位を次の表に示す。

[8] http://oauth.rubyforge.org/
[9] http://mecab.sourceforge.net/

表 5.1: 名詞の出現回数上位 10 個

名詞	出現回数
の	9032
ん	7722
人	4387
こと	4287
さん	2840
今日	2488
これ	2120
何	2018
日	2018
そう	2018

この表をみてわかるように、単純にツイートを形態素に分解するだけでは、明らかにトピックを定義するのに不適である名詞ばかりが、上位にくることがわかった。

これは、Twitter における大多数のユーザは、必ずしもきれいな日本語を用いようという意識が無いために、非常に砕けた日本語の使い方をする場合が多いが、その一方で、今回使用した形態素解析エンジンでは整った日本語の使い方を前提としたものであったからだと考えられる。

この問題と、後述の手法で十分にトピックの抽出が可能なことにより、この名詞によるトピック抽出は、この実装においては用い無いこととした。

名詞によってトピックを定義するためには、不自然言語処理などと呼ばれる、Twitter などの砕けた日本語の形態素解析のノウハウがさらに必要であると考えられる。

ハッシュタグによるトピックとそれに対するユーザの貢献度の決定

Twitter では、共通の話題について話す時に、ハッシュタグ (図 4.1) というものを用いることが一般的である。したがって、Twitter において、ハッシュタグをつけられたツイートは、そのハッシュタグによってまとめられるトピックに属するツイートであると考えることができる。今回の実装ではこれを利用し、ハッシュタグ付きのツイートに着目して、トピックに分類した。

処理の流れを以下に示す。

1. まず収集したツイートに存在する全ハッシュタグの出現回数を数える。
2. 発見されたハッシュタグを出現回数順にソートする。
3. ハッシュタグを含む各ツイートは、ハッシュタグによって定まるトピックに属すると考えて、各ユーザのツイートをハッシュタグごとにまとめる。

4. ユーザのハッシュタグごとにまとめられたツイートを、更に単位時間ごとに区切ってツイート数を数える。

本実装においては、ユーザがトピックに属するツイートを何度行ったかによってユーザの貢献度を定めた。したがって、以上の処理により各単位時間ごとの各ユーザのトピックへの貢献度が定まる。

5.3.2 ユーザ描画位置の決定

ユーザの描画位置の決定のためには、まずユーザ同士の性質の近さを考える必要がある。

本実装においては、各ユーザがそれぞれのトピックにどの程度属しているか、といった情報をユーザの性質と考える。すなわち、ユーザの性質を表すベクトル $\vec{u}$ を、

$$\vec{u} = (t_0, t_1, \cdots, t_n)(t \text{ はトピックへの貢献度}, n \text{ は全トピック数})$$

と定義した。続いて、本実装においてはユーザ同士の性質の近さは、コサイン類似度を用いた。すなわち、類似度 $s(0 \leq s \leq 1)$ を

$$s_{ij} = \frac{\vec{u_i} \cdot \vec{u_j}}{\|\vec{u_i}\|\|\vec{u_j}\|}(\vec{u_i}, \vec{u_j} \text{はそれぞれユーザを表すベクトル})$$

のように計算した。

多次元尺度構成法

コサイン類似度を用いて算出された、ユーザ同士の類似度をもとにして、対象とする全ユーザの類似度行列 A は

$$A = \begin{pmatrix} s_{00} & s_{01} & \cdots & s_{0k} \\ s_{10} & s_{11} & \cdots & s_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{l0} & s_{l1} & \cdots & s_{lk} \end{pmatrix}$$

となる。この行列 A を多次元尺度構成法を用いて 2 次元に圧縮し、これを x, y 平面の座標として用いる。

5.4 描画

5.4.1 HTML5/Javascript

描画のためのクライアント環境は、HTML5 を使うことが出来るブラウザを想定しており、また、実際の描画のプログラムは、HTML5 の canvas タグと Javascript を用いて作成した。描

画クライアントは、canvas を用いた描画用のフレームワークである VSFx[10]を活用した。

5.4.2 データの取得と描画

先の手順までで、各トピックの広がり方を可視化するために必要なデータが得られているので、クライアント側では、サーバからのそのデータを受け取って、そのまま描画を行う。

[10]https://www.assembla.com/code/vsfx/subversion/nodes

第6章　ユースケース

6.1　実行例

Twitter において、明らかに筑波大生であると判断出来る 630 ユーザを対象として、2011 年 12 月 09 日 22 時 51 分 52 秒から 2011 年 12 月 20 日 20 時 26 分 02 秒までの間で、111,647 ツイートを継続的に取得した。

6.1.1　ランダムな配置と多次元尺構成法による配置

描画位置決定については、ランダムに配置したものと、多次元尺度構成方により配置したものの両方を試した。

先ず、以下の図 6.1 が、ランダムに配置したものである。

一方、次の図 6.2 が、ユーザの類似度を計算して、それをもとに多次元尺度構成法を用いて配置したものである。

これを見てわかるように、多次元尺度構成法を用いたものは全体的に真ん中に引き寄せられている。

また、次の拡大図 6.3 を見てわかるように、ピンクの境界線がついたものは右上部分に集まっている、黄色の境界線がついたものは左下に集まっているというように、同じトピックに参加するユーザは近くに配置されていることが観察出来る。

ランダムな配置に比べて、コサイン類似度と多次元尺度構成法を用いた配置のほうが、同じトピックに属するユーザ同士が近く配置されるようになり、トピックに属するユーザがわかりやすく表現出来ている。

6.1.2　アニメーションの観察

描画されたアニメーションを再生すると、拡大図 6.3 の真ん中付近に観察できる水色の境界線がついた複数のユーザで、興味深い動きが観察できた。以下の図 6.4、図 6.5、図 6.6 は、その部分のみを抜粋したものである。

図 6.4 から始まり、図 6.5、図 6.6 へと変化する様子が観察できた。このトピックはハッシュタグ「#spcamp」がついたものであった。

最初の図 6.4 の状態ではあまり活発では無かったが、図 6.5 になると、右上と左下のユーザが活発になった。そして、最後には図 6.6 に見られるように、右下のユーザと左上のユーザを巻き込んでいた。

図 6.1: ランダムに配置した例

図 6.2: 多次元尺度構成法を用いて配置した例

図 6.3: 多次元尺度構成法を用いて配置した例の拡大

この動きが観察できたハッシュタグ「#spcamp」について調べてみると、12 月 17 日に行われたイベントに関するつぶやきで、図 6.5 の時点で活発であった 2 ユーザは、実際にイベントに参加していた。しかし、その後の図 6.6 の時点のトピックで活発さを増したユーザは、イベントに参加しておらず、Twitter 上で、先の 2 ユーザのつぶやきを見て、トピックに参加した様子が発見できた。

このように、小規模ではあるが、確かにトピックの広がる様子を観察することができた。

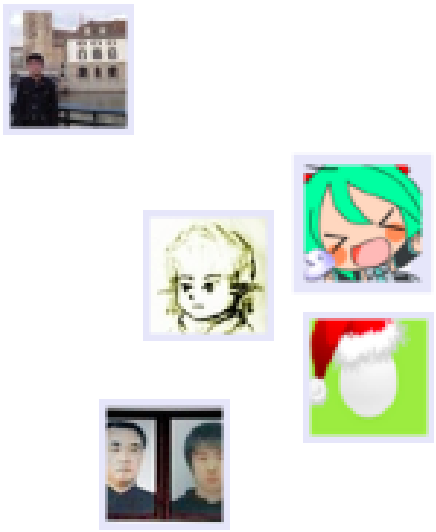

図 6.4: トピックの広がる様子 1 (図 6.2 をアニメーションしたものの抜粋)

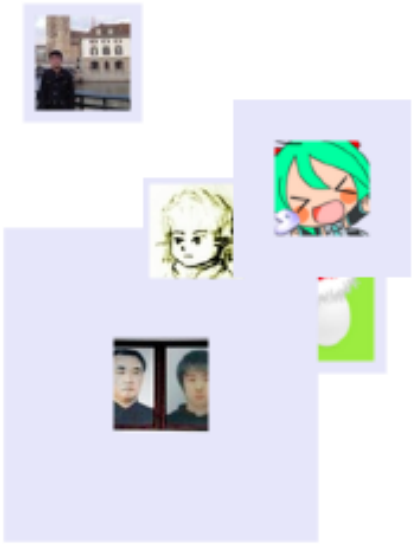

図 6.5: トピックの広がる様子 2 (図 6.2 をアニメーションしたものの抜粋)

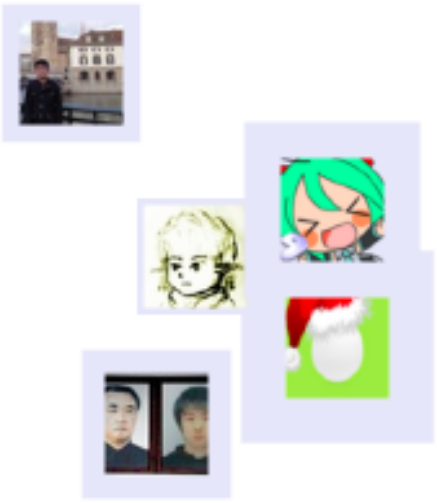

図 6.6: トピックの広がる様子 3 (図 6.2 をアニメーションしたものの抜粋)

第7章　結論

本研究では、マイクロブログにおいて、あるユーザとそのユーザがフォローしているユーザによって形成されるコミュニティにおいて、トピックがどのように広がるかを直感的に観察できるようにするための手法を提案した。この提案手法には、ツイートを収集してそこからトピックを抽出、更にはユーザのトピックへの貢献度の時間変化を考えて、最終的にユーザがトピックの広がる様子を直感的に観察出来るようにする可視化までのプロセスを含む。

そして、この手法の実装例を作成した。この実装では、Twitter におけるトピックの抽出方法については、ハッシュタグを用いた。更に、描画位置決定の手法としては、ユーザがどのトピックにどれだけ参加するかによるユーザの各トピックへの貢献度によって、ユーザの性質を表すベクトルを定め、このベクトル間の類似度をコサイン類似度を用いて算出してユーザ間の類似度を定め、そして更にこれを多次元尺度構成法を用いて 2 次元に圧縮して描画位置を決定するという手法を実装した。

実際に、実装例を用いて、筑波大生 630 ユーザの 2011 年 12 月 09 日 22 時 51 分 52 秒から 2011 年 12 月 20 日 20 時 26 分 02 秒の間のツイートを継続的に収集し、この間のトピックの観察を試みた。小規模ではあったが、実際に 12 月 17 日のあるイベントに関するトピックが広がる様子などを直感的に観察することが出来た。また、コサイン類似度と多次元尺度構成法を用いての描画位置決定の有効性も確認できた。

今後の発展として、よりよいトピックの抽出法を開発して本手法に適用する、もしくは、よりよい描画位置決定方法を開発して本手法に適用するなどの発展を考える。また、色の選択についても、トピックの意味的な内容に即した色を選択するなどの発展が考えられる。

謝辞

本研究を行うに当たって、指導教員である三末和男准教授を初めとして、田中二郎教授、志築文太郎講師、高橋伸准教授には丁寧なご指導や貴重なご意見を頂きました。心から感謝申し上げます。また、インタラクティブプログラミング研究室の先輩のみなさま、同学年のみなさまにも多くのご意見やご助言を頂きまして大変感謝しております。特に NAIS チームのみなさまには、日頃の研究活動やゼミの発表、議論など様々な場で、ご指導ご鞭撻頂きましたこと深く感謝しております。また、大学生活 4 年間において公私ともに支えて頂きました、多くの友人や先生方にも厚く御礼申し上げます。最後に、学生生活を送る上で様々な面から多大なる援助を頂きました、家族には大変感謝しております。学生生活で御世話になりました全てのみなさまに、深く感謝申し上げます。本当に有り難うございました。

参考文献

[1] D. Ramage, S. Dumais, and D. Liebling. Characterizing Microblogs with Topic Models. ICWSM ’10, AAAI Press, pp. 130-137, (2010).

[2] 数原良彦, 戸田浩之, 櫻井彰人. ブログにおけるイベントマイニングのための適切なキーワード抽出. DEWS ’07, A2-6, (2007).

[3] M. Cataldi, L. D. Caro, and C. Schifanella. Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation. MDMKDD ’10, ACM, 10 pages, (2010).

[4] M. S. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. H. Chi. Eddi: Interactive Topic-based Browsing of Social Status Streams. UIST ’10, ACM Press, pp. 303-312, (2010).

[5] A. Java, X. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. WebKDD ’07, ACM Press, pp. 56-65, (2007).

[6] M. Naaman, J. Boase and C. Lai, Is it Really About Me? Message Content in Social Awareness Streams. CSCW ’10, ACM Press, pp. 189-192, (2010).

[7] C. Honeycutt and S. C. Herring. Beyond Microblogging: Conversation and Collaboration via Twitter. HICSS ’09, IEEE (2009), 10 pages.

[8] J. Zhang, Y. Qu, J. Cody and Y. Wu. A case study of micro-blogging in the enterprise: use, value, and related issues. CHI ’10, ACM Press, pp. 243-252, (2010).

[9] Ehrlich, K. and Shami, N. Microblogging Inside and Outside the Workplace. ICWSM ’10, AAAI Press, pp. 42-49, (2010)

[10] J. Leskovec, L. Backstrom and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. KDD ’09, ACM Press, pp. 497-506, (2009).

[11] D. A. Shamma, L. Kennedy and E. F. Churchill. Tweet the debates: understanding community annotation of uncollected sources. WSM ’09, ACM Press, pp. 3-10, (2009).

[12] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. SIGIR ’08, ACM Press, pp. 491-498, (2008).

[13] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. EMNLP '03, ACL , pp. 216-223, (2003).