

SNS 経由で入手される情報のユーザ間差異の可視化

プロジェクトマネジメントコース

ソフトウェア開発管理グループ

矢吹研究室

1242131

吉野 聡志

謝辭

目次

第 1 章	序論	7
1.1	論文構成	7
第 2 章	背景	9
第 3 章	目的	11
3.1	目的	11
3.2	プロジェクトマネジメントとの関連性	11
第 4 章	手法	13
4.1	本章の構成	13
4.2	Twitter とは	13
4.3	用語	14
4.4	Twitter API について	16
4.5	Oracle VM VirtualBox とは	17
4.6	用語	17
4.7	研究の手法	18
第 5 章	結果	27
第 6 章	考察	29
第 7 章	結論	31
	参考文献	33

第 1 章

序論

1.1 論文構成

本稿では以下のような構成をとる．．．．

第 2 章

背景

世界的に人気のある SNS(Social Networking Service) のひとつとして Twitter が存在する．2015 年 6 月 30 日現在，月間アクティブユーザは 3 億 1600 万人である．SNS の中でもアクティブユーザ数が非常に多く，利用スタイルも多数ある Twitter に対し，ユーザである人々が顕在的・潜在的に持っているニーズが何であるかが分かれば，他の SNS（Facebook 等）との差別化を図りやすくなる．これにより効率的なマーケティングの手法を Twitter 社や，Twitter 上に広告を打ち出す企業に提案できるのではないかと考えられる．

第 3 章

目的

3.1 目的

後述する方法で数名の Twitter タイムラインを取得し，各人のタイムライン上に並ぶ単語や，単語同士の結びつきの強さを可視化する．その結果からつぶやきの性質を分析し，各人の嗜好や関心事項と合致するもの・しないものを見つけ出し，Twitter への顕在的・潜在的なニーズを読み取ることが目的である．

3.2 プロジェクトマネジメントとの関連性

...

第 4 章

手法

4.1 本章の構成

本章では、本研究で利用する Twitter の機能や、研究を行うための環境設定・利用するサービスについて解説したのち、研究の手法について記す。

4.2 Twitter とは

Twitter は、つぶやきと呼ばれる短い発言を投稿し、他のユーザが閲覧・返信することによってコミュニケーションが生まれる SNS である。他のユーザのつぶやきを追跡することを「フォローする」という。タイムラインと呼ばれる画面には、自分のつぶやきとフォローしたユーザのつぶやきが同一画面上に、リアルタイムで表示される。相手がつぶやきを非公開にしていなければ、フォローをする際の承認は不要で、つぶやきを引用することもできる。つぶやきやユーザの検索をすることも可能で、ユーザ同士がつながる機会が多岐にわたる。



図 4.1 日本語版 Twitter 公式アカウントのプロフィール画面

4.3 用語

本節では、本研究で利用する Twitter の用語について説明する [1]。

4.3.1 スクリーンネーム

ユーザがアカウントを登録する際、ユーザ同士を区別するために設定する名前。最大 15 文字の半角英数字とアンダースコアの組み合わせからなる。スクリーンネームの先頭には@が付く。登録した後、ユーザによる変更が可能。

4.3.2 ユーザ ID

ユーザがアカウントを登録する際、Twitter がユーザを識別するために決定する数字の羅列。スクリーンネームとは異なり、登録した後の変更は不可能。

4.3.3 名前

ユーザのプロフィールページでスクリーンネームとともに表示される名前。スクリーンネームと違い、日本語や記号を入れることができ、他のユーザとの重複があってもそのまま登録できる。登録後のユーザによる変更が可能。

4.3.4 ツイート

Twitter における「つぶやき」のこと。1 つのツイートにつき 160 文字まで、という制限が設けられている。ただし、そのうちの 20 文字はユーザ ID に割り当てられているため、残りの 140 文字がツイートの本文ということになる。

4.3.5 タイムライン

自分とフォローしているユーザのツイートがリアルタイムに、時系列に沿って表示される画面。この画面が Twitter では基本となるため、「ホーム」とも呼ぶ。

4.3.6 フォロー

特定のユーザのツイートをタイムラインに表示するための操作。

4.3.7 フォロワー

特定のユーザをフォローしているユーザを指す。ユーザ A がユーザ B をフォローしている場合、ユーザ A はユーザ B のフォロワーである。

4.3.8 リプライ（返信）

特定のスクリーンネームから始まるツイート。そのユーザ宛のツイートということになる。

4.3.9 リツイート

他のユーザのツイートを、自分のフォロワーに向けてツイートする操作。

4.3.10 お気に入り

他のユーザのツイートを、自分のお気に入りリストへ登録する操作。

4.3.11 メンション

特定のスクリーンネームを含むツイート。リプライと違い、ツイート内のどこにスクリーンネームが入っていてもメンションとなる。非公式クライアントの種類によっては自分のスクリーンネームが入ったメンションを表示する機能がある。

4.3.12 アクティビティ

自分がフォローしているユーザが他のユーザをフォローしたり、ツイートをお気に入りに登録した等の活動履歴を一覧で表示するページ。フォロー相手の行動をリアルタイムで確認できるため、タイムラインの閲覧だけでは得られない新たな発見の可能性を生み出している。

4.3.13 リスト

ユーザをグループにまとめ、管理する機能。リストごとにタイムラインを表示させることができる。

4.3.14 ハッシュタグ

ツイート内に記載された#から始まる文字列．ハッシュタグを付けてツイートするとその部分が自動的にリンクとなり，クリックすると同じタグが付いたツイートの検索結果が時系列で表示される．

4.3.15 トレンド

Twitter 上のツイートの中で多く話題に上っているキーワードをリアルタイムに抽出し，表示する機能．表示するトレンドの地域を設定でき，全世界のトレンドを表示することもできる．

4.3.16 短縮 URL

ツイートの文字数を URL 含め 140 字以内に抑えるため，長い URL を短い URL に変換したもの．代表的なものとして Twitter 公式 (<http://t.co>) がある．

4.3.17 非公開ツイート

フォロワーにのみツイートを公開する設定．非公開ツイートを設定しているユーザのツイートを閲覧するには，その相手に対しフォローリクエストを送信し，許可を得ることが必要となる．

4.3.18 認証済みアカウント

有名人等，多くの人に検索されたり，なりすましの対象になりうるユーザの Twitter アカウントについて，Twitter 社が本人確認を行ったアカウント．認証済みアカウントにはプロフィール画面等にチェックマークの入ったブルーの認証バッジが表示される．

4.4 Twitter API について

本研究では Twitter から様々な情報を取得するためのプログラムを使用する．その際に必要となるのが Twitter API である．本節では，Twitter API に関する説明をする．

4.4.1 API

アプリケーションプログラムインターフェイス (Application Program Interface) の略で，プログラミングの際に使用できる命令や規約，関数等の集合・枠組みのこと．ソフトウェア開発の際，一からすべてを作るには膨大な時間を要するが，API という枠組みを利用することにより，プログラミングの負担を軽減することができる [2]．

4.4.2 Twitter API

Twitter 社が提供する API．これを利用することで Web サイトやアプリケーションなどから Twitter の機能呼び出すことができ，ツイートの参照や検索等を行えるアプリケーション開発を行えるようになる [3]．

4.4.3 REST API

Twitter API のパラメータ (リソース) を指定し、特定の URL に HTTP でアクセスすると、JSON 形式で記述されたメッセージがレスポンスされるシステムのこと。これはツイートの更新や参照を行う際に使用する基本的な API となる。ただし利用制限があり、15 分以内に同じ機能を特定の回数利用すると、はじめにその機能を利用した時間から 15 分経過するまではその機能が利用できなくなる。

4.4.4 Streaming API

今回利用する API。タイムラインの変更をリアルタイムに受け取ることができる。REST API と異なり、利用制限は設けられていない。

4.4.5 Access Token

Twitter に限らず、多くのユーザアカウントに存在する ID とパスワードのような「Access Token」と「Access Token Secret」の 2 種類の文字列を指す。これは、アプリケーションがユーザの代わりにユーザデータにアクセスするための「通行許可証」のようなものといえる。悪意ある第三者に ID とパスワードそのものを預けた場合、勝手にログインされたり、最悪の場合アカウントを削除することも可能となってしまう。そういったリスクを防ぐため、限られた範囲でユーザデータにアクセスできる権限として Access Token が利用される [4]。

4.5 Oracle VM VirtualBox とは

Oracle VM VirtualBox (以下、VirtualBox と表記する) は、使用している PC マシン上に仮想的なマシンを作成し、別の OS をインストール・実行することができるオープンソースソフトウェアである。Windows や Mac OS X, Linux 等、様々な OS で利用することができる [5]。

4.6 用語

本節では、本研究で利用する VirtualBox の用語について説明する。

4.6.1 ホストマシン (物理マシン)

物理的に存在するコンピュータ。

4.6.2 ホスト OS

ホストマシンにインストールされている OS。VirtualBox はホスト OS にインストールされる。本研究で使用するホストマシンの OS は Windows 7/8.1 である。

4.6.3 バーチャルマシン (仮想マシン)

VirtualBox が作成する論理的なマシン。VirtualBox がホストマシンのコンピュータ資源 (CPU やメモリ, HDD 等) の一部を仮想化し、ゲストマシンに割り当てる。ホストマシンの資源を使いきらない限り、ゲストマシンを複数作成したり、多重起動させることができる。

4.6.4 ゲスト OS (仮想 OS)

ゲストマシンにインストールされる OS . 本研究では , Linux ディストリビューションの一つである Ubuntu をインストールした .

4.6.5 仮想ディスク

ゲストマシンが使用する仮想のハードディスク . パーチャルマシンからはこれを物理ディスクとして扱うことができる . 仮想ディスクの実体はホストマシン内にファイルとして存在する .

4.6.6 キャプチャ

Guest Additions (後述) をインストールしていない状態でゲスト OS の画面をクリックした際 , キーボードやマウスの入力ゲスト OS の中でしかできなくなってしまう状態 . ホストキー (初期状態では右側の Ctrl キー) を押す度に入力先をホスト OS とゲスト OS に切り替えられるが , その手間を省く等の理由で Guest Additions をインストールする .

4.6.7 Guest Additions

VirtualBox の操作性を向上させるためのモジュール . ホスト OS とゲスト OS 間のシームレスなマウスポインタの移動や , ゲスト OS の画面の解像度を自由に変更することなどが可能となる [6] . これはゲスト OS にインストールするため , ゲストマシンごとにインストール作業をする必要がある .

4.7 研究の手法

VirtualBox のインストール

<http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html> よりインストーラをダウンロードし , インストーラの指示に従って VirtualBox をインストールする .

Ubuntu のインストール

まず , <https://www.ubuntulinux.jp/download/ja-remix> より Ubuntu 14.04 の ISO イメージをダウンロードする . 本研究では 64bit 版を選択した .

次にインストールした VirtualBox を立ち上げ , ウィンドウ左上にある「新規 (N)」のボタンを押してゲストマシンの作成を行う .

ゲストマシンの名前 , メモリサイズ , HDD の設定をするとウィンドウが閉じられる .

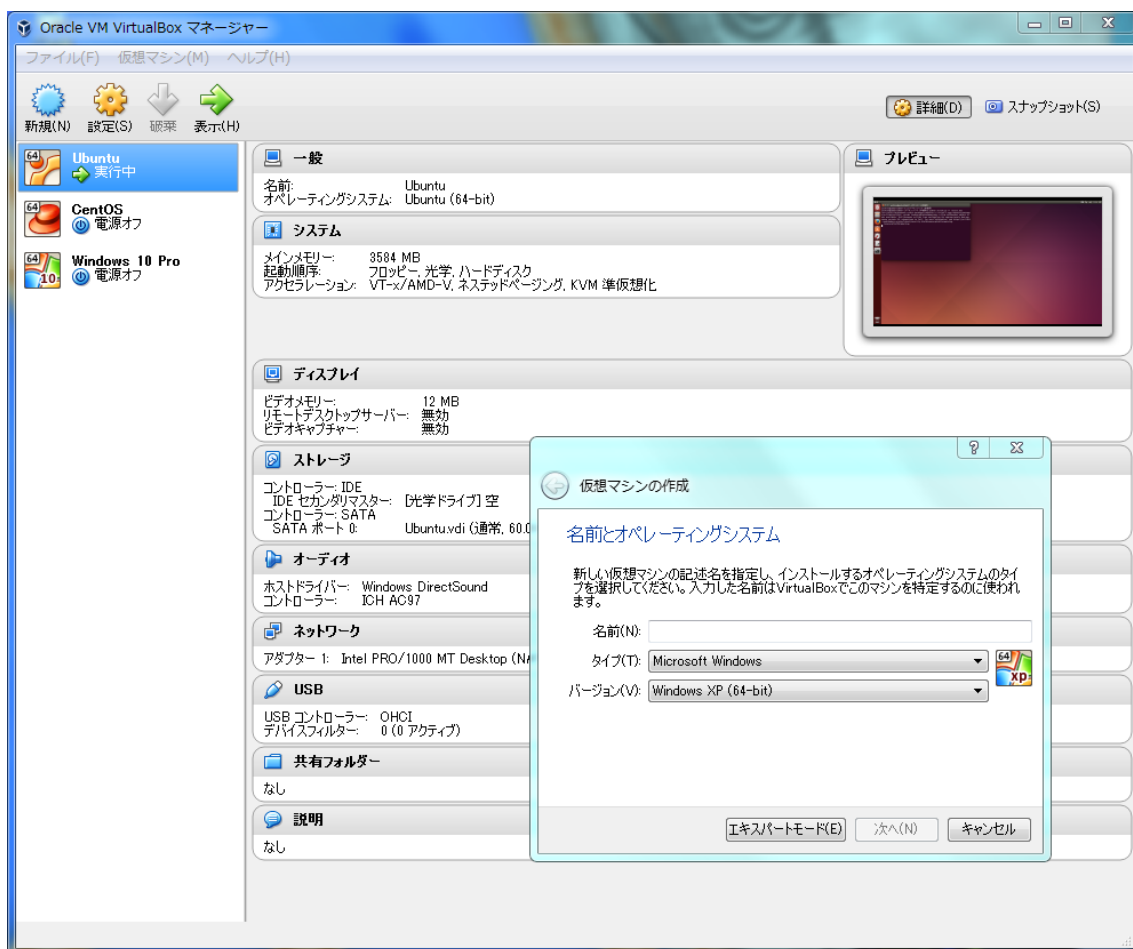


図 4.2 VirtualBox のウィンドウとゲストマシン作成ウィンドウ

続いて「設定 (S)」を開き、「ストレージ」の項目にあるストレージツリーより、「コントローラー: IDE」内の「空」を選択する。

「属性」内にあるディスクのマークをクリックしてプルダウンメニューを開き、「仮想光学ディスクファイルを選択...」をクリックしてダウンロードした ISO イメージをマウントする。

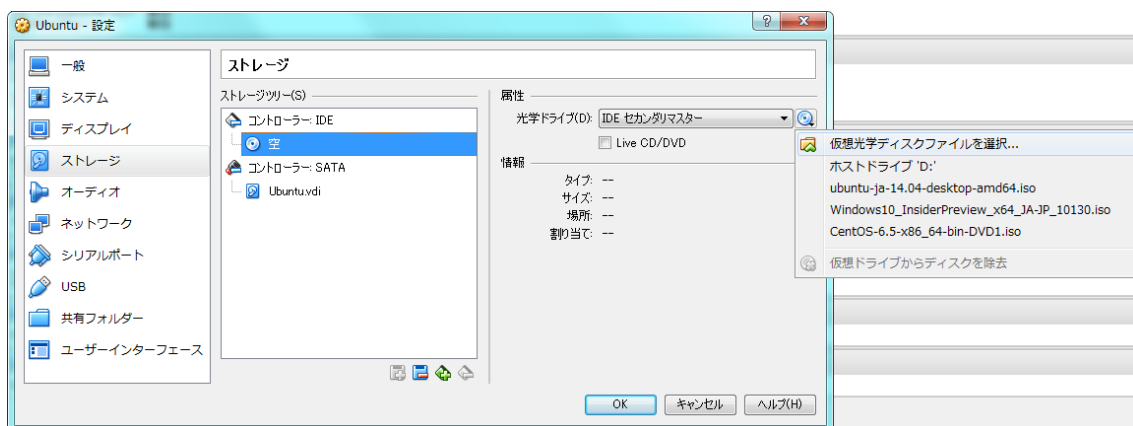


図 4.3 ISO イメージのマウント

「OK」をクリックして「起動 (T)」を押すとゲストマシンが起動し、Ubuntu のインストールウィザードが表示される。

表示内容に従ってウィザードを進めていくと、ゲストマシン内に Ubuntu がインストールされる。

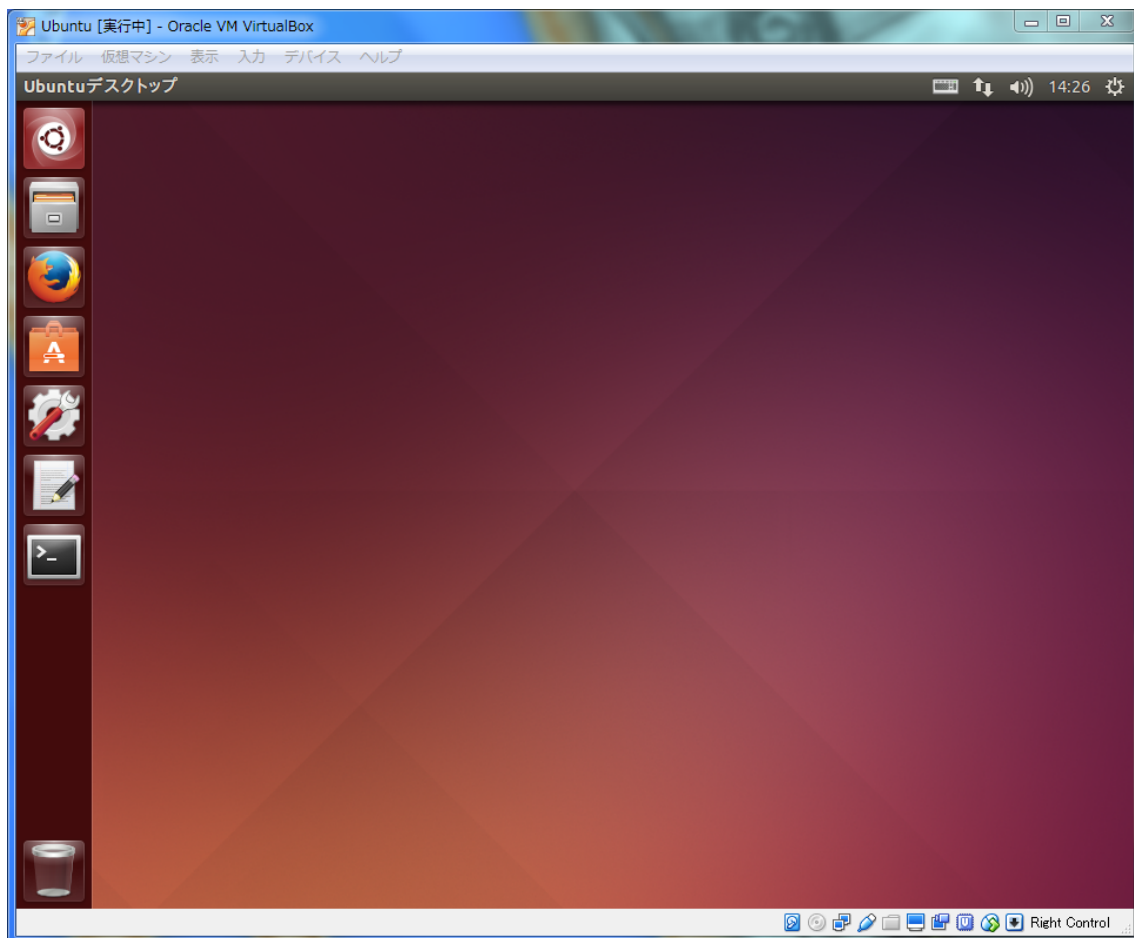


図 4.4 VirtualBox 上で動作する Ubuntu 14.04

Guest Additions のインストール

ゲストマシンにインストールした Ubuntu を立ち上げ、ホスト OS 側のメニューバーにある「デバイス」から「Guest Additions CD イメージの挿入...」を選択するとウィンドウが開く。

「実行する (R)」を押してインストールし、Ubuntu を再起動すると Guest Additions の機能が有効になる。

Tweepy のインストール

本研究では、Python というプログラム言語で Twitter を利用するためのライブラリである Tweepy をインストールした。

Ubuntu のアプリケーション一覧より「端末」を開き、下記のコマンドを実行すると Tweepy がインストールされる。

```
1 sudo apt-get install python-setuptools python-pip
2 sudo easy_install tweepy
```

タイムラインの取得

本研究では、矢吹研究室に所属する 3 年生のうち、Twitter のアクティブユーザである 4 人の協力を得て、各々のマシンや Twitter アカウントを利用して Twitter タイムラインを取得した。

所属する研究室の指導教員が自らのブログに記載した方法に則り、Streaming API を利用して Twitter のタイムラインを取得する。方法は下記のとおりである。

Twitter にログインし、画面右上のアカウントアイコンをクリックするとプルダウンメニューが表示さ

れるため、「設定」を選択する。

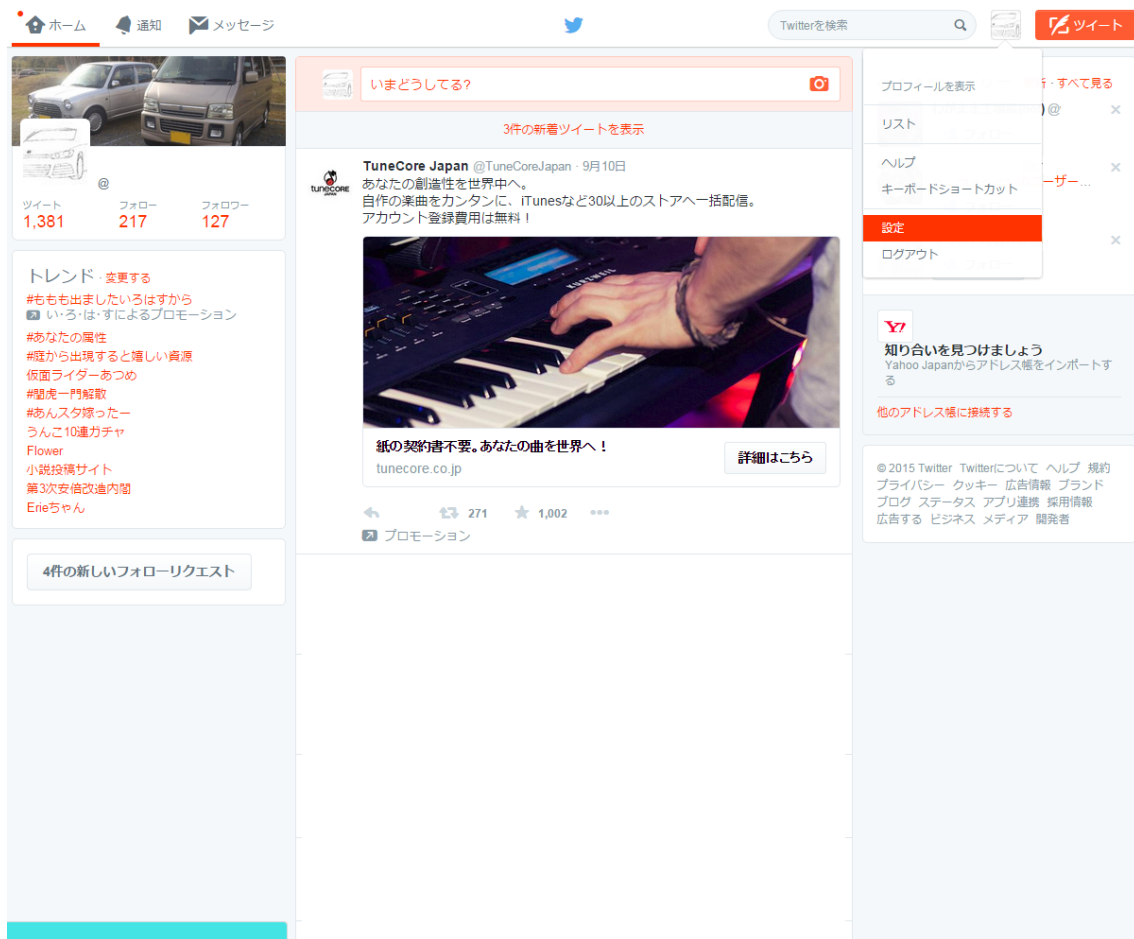


図 4.5 プルダウンメニューの表示

左側のメニューより「モバイル」を選択し、携帯電話番号を入力して「続ける」をクリックする。



図 4.6 携帯電話番号の入力画面

Twitter の開発者向けページ (<https://dev.twitter.com>) にアクセスし、画面下方にある「Manage Your Apps」(下図の赤枠で囲った部分) をクリックする。

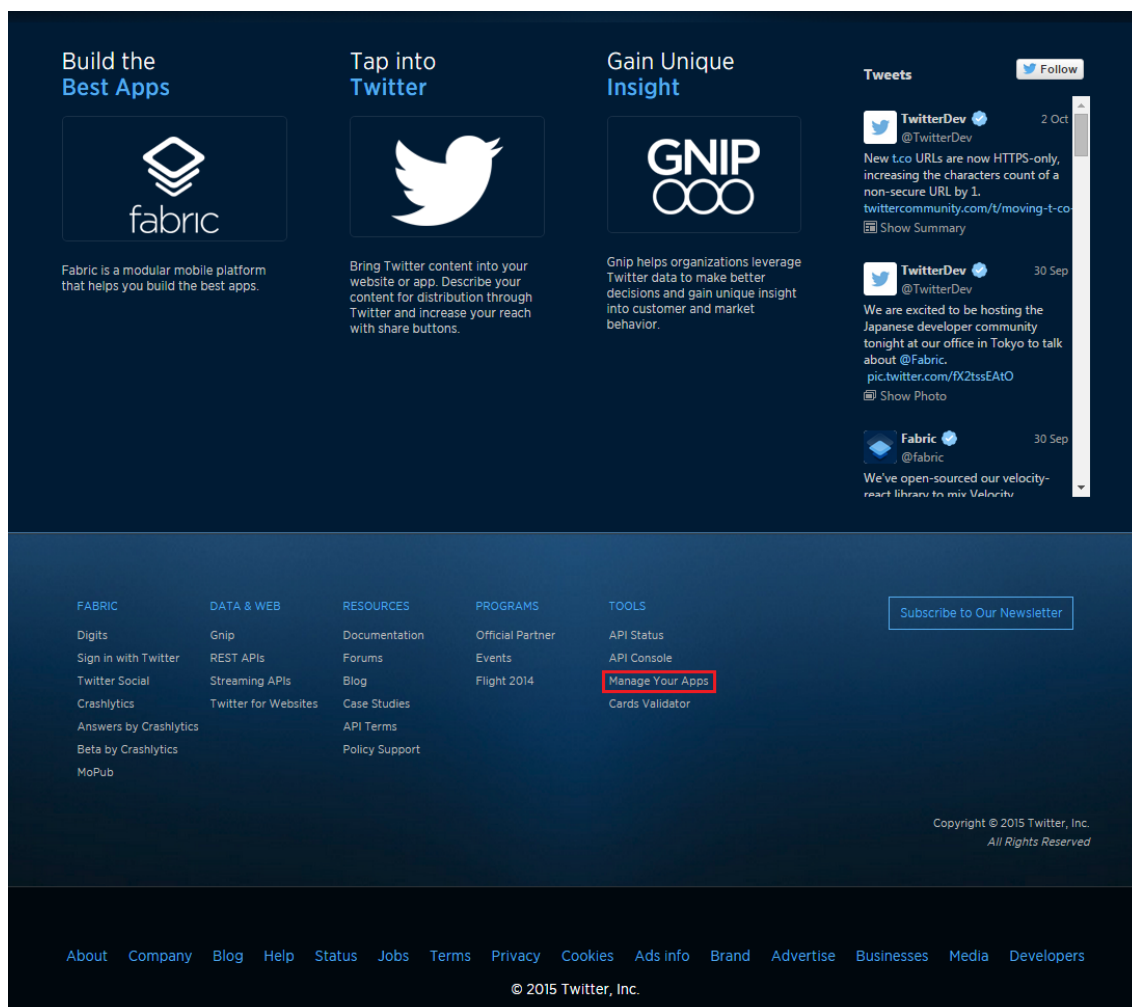


図 4.7 Twitter の開発者向けページ

「You don't currently have any Twitter Apps.」と表示されるので、その下にある「Create New App」をクリックし、アプリケーション作成画面に移る。

「Name (アプリケーションの名前)」、「Description (アプリケーションの説明)」、「Website (アプリケーションを動かす Web サイトのトップページの URL を本来入力しなければならないが、そういった Web サイトがないため、ここでは各 Twitter アカウントのプロフィールページの URL を記載しておく)」をそれぞれ入力し、規約に同意して「Create your Twitter application」をクリックするとアプリケーションが作成される。

これにより、後述するプログラムに必要な情報 (Consumer Key, Consumer Secret, Access Token, Access Token Secret) を取得することができる。

ただし、Access Token と Access Token Secret に関してはアカウントごとに作成する必要があるため、それらを作成するための Web サービスを用いる。

http://getaccesstoken.herokuapp.com/にアクセスし、表示されたページの指示に従って作成したアプリケーションの設定を変更する。

Access token を取得できるツールです。BOTの作成などにご利用ください。

このツールを使う前に <http://twitter.com/apps> に登録したアプリケーションの設定から Application TypeをBrowserに変更して、Callback URLに適当なURL(<http://localhost/> や <http://getaccesstoken.yayugu.net/> など)を登録しておいてください。

詳しくは<http://d.hatena.ne.jp/yayugu/20100908/1283953165>をご覧ください

Consumer key:

Consumer secret:

author: [@yayugu](#)

図 4.8 <http://getaccesstoken.herokuapp.com/>の表示

変更が済んだら、ブラウザで予め Access Token と Access Token Secret を取得したいアカウントでログインし、再び上記サイトにアクセスする。

「Consumer key」、「Consumer secret」に自分のアカウントのそれぞれの情報を入力し、「get access_token」をクリックすると Access Token と Access Token Secret が表示されるため、これらをメモしておく。

「端末」に「python stream.py » result.dat」と入力して下記のプログラム（stream.py）を実行する（consumer_key と consumer_secret はアプリケーション作成時に取得した自分のアカウントのそれぞれの情報を、access_token と access_token_secret は各々のアカウントの情報を予め入力しておく）。

```

1      # -*- coding: utf-8 -*-
2
3      from tweepy.streaming import StreamListener
4      from tweepy import OAuthHandler
5      from tweepy import Stream
6
7      consumer_key = ""
8      consumer_secret = ""
9
10     access_token = ""
11     access_token_secret = ""
12
13     class StdOutListener(StreamListener):
14         def on_data(self, data):
15             if data.startswith("{"):
16                 print data
17             return True
18
19         def on_error(self, status):
20             print status
21
22     if __name__ == '__main__':
23         l = StdOutListener()
24         auth = OAuthHandler(consumer_key, consumer_secret)
25         auth.set_access_token(access_token, access_token_secret)
26
27         stream = Stream(auth, l)

```



```
28 stream.userstream()
```

取得したタイムラインの処理

取得したタイムラインは JSON 形式で保存されており (result.dat というファイルに保存されている), この形式のままではつぶやきを閲覧したり, 分析することができない.

そこで下記の parse.py を用いて必要なデータだけを, 正しく読める形にして抽出する.

「端末」に「python parse.py 20151005000000 20151005235959 < result.dat」と入力して実行すると, 日本時間の 2015 年 10 月 5 日 00 時 00 分 00 秒から 2015 年 10 月 5 日 23 時 59 分 59 秒のつぶやきの本文のみが抽出され, 「端末」上に表示される.

parse.py のソースコードは以下のとおりである.

```
1  #!/usr/bin/env python
2  import sys, json, time, calendar
3  #from pprint import pprint
4
5  def YmdHMS(created_at):
6      time_utc = time.strptime(created_at, '%a %b %d %H:%M:%S +0000
7          %Y')
8      unix_time = calendar.timegm(time_utc)
9      time_local = time.localtime(unix_time)
10     return int(time.strftime("%Y%m%d%H%M%S", time_local))
11
12     argv = sys.argv
13     start_time = 0
14     end_time = 999999999999999
15     if 1 < len(argv):
16         start_time = int(argv[1])
17         end_time = int(argv[2])
18
19     for line in sys.stdin:
20         try:
21             tweet = json.loads(line)
22             #pprint(tweet)
23             tweet_time = YmdHMS(tweet['created_at'])
24             if start_time <= tweet_time and tweet_time <=
25                 end_time:
26                 tweet_sec = tweet_time - start_time
27                 screen_name = tweet['user']['screen_name']
28                 text = tweet['text'].encode('utf-8')
29                 url = "https://twitter.com/#!/%s/status/%s"\
30                     % (screen_name, tweet['id_str'])
31                 #print tweet_sec, url, text
32                 #print text
33                 t = time.strptime(str(tweet_time), "%Y%m%d%H
34                     %M%S")
```

```
32         print time.strftime("%H:%M:%S", t), text
33     except StandardError:
34         pass
```

タイムラインの分析

上記手順で抽出したつぶやきはテキストエディターにコピー&ペーストし、テキストファイルとして保存する。

UserLocal テキストマイニング (<http://textmining.userlocal.jp/>) にアクセスし、「テキストファイルを解析」の「解析ページへ」をクリックする。

保存したテキストファイルを選択し、「解析する」をクリックすると解析結果が表示される。なお、一度に解析できる文字数は 100,000 文字までである。

第 5 章

結果

第 6 章

考察

第 7 章

結論

参考文献

- [1] Digital Garage, Inc. ツイッター (twitter) 用語集 - ツイナビ. <http://twinavi.jp/guide/section/twitter/glossary> (参照 2015-10-06).
- [2] IT Trend. Api とは - 意味の解説 | it トレンドの it 用語集. <http://it-trend.jp/words/api> (参照 2015-10-06).
- [3] ディレクター F. やってみると意外とカンタン! twitterapi を使ってつぶやきデータを取得してみた。| web 制作なら webnaut. <http://webnaut.jp/develop/633.html> (参照 2015-10-06).
- [4] Yuta Arai. Twitter api の使い方まとめ. <https://syncer.jp/twitter-api-matome> (参照 2015-10-07).
- [5] IT Engineer Sideline. 用語の解説 | virtualbox mania. <http://vboxmania.net/term> (参照 2015-10-05).
- [6] SoSO Corporation. Linuxmania: Virtualbox を使おう - hint & tips (ubuntu). http://www.linuxmania.jp/virtualbox_02.html (参照 2015-10-05).