デジタルプラクティス　Vol.11 No.1(Jan. 2020)

推薦投稿論文

大量ツイートの収集・分析を個人で手軽に実現可能にする方法の提案

松浦 智之[1]　當仲 寛哲[1]　大野 浩之[2]

[1]（有）ユニバーサル・シェル・プログラミング研究所　[2]金沢大学総合メディア基盤センター／慶應義塾大学SFC研究所

短文投稿SNS“Twitter”は今や多くの人々に認知され，NHKを始めとした日々のニュース番組等においても，もはやTwitterやツイート（Twitterに投稿される文章）が何かという説明が省略されながら，世論を反映した情報源として引用あるいは分析されている．しかしながら，社会現象のような膨大な量のツイートを発生させる話題を分析しようとなると，すでによく知られている方法では費用的にも技術的にも個人には敷居が高い．本稿では，一定の制約はある中でも，個人による大量ツイートデータの収集・分析を実現し得る手法を提案し，実際に，日本国内で社会現象を起こして大量のツイートを発生させた2つの話題に関するツイートの収集・分析を行うことで，提案手法の実用性を示している．

★本稿の内容は2018年年7月6日のDICOMO 2018シンポジウムにて報告され，IOT研究会主査により情報処理学会デジタルプラクティスへの掲載が推薦された論文である．

1. はじめに

短文投稿SNS“Twitter”は今や多くの人々に認知され，NHKを始めとした日々のニュース番組等においても，もはやTwitterやツイート（Twitterに投稿される文章）が何かという説明が省略されながら，世論を反映した情報源として引用あるいは分析されている．

ツイートの分析は，そのような大きな組織のみならず，個人やそれに類する小規模な研究チームにとっても有用である．しかし，世界中で日々投稿されるツイートデータの量は膨大で，それらをすべて，あるいはリアルタイムに取得するために提供されているAPIの利用料は非常に高額である．また，データが膨大であるために，それらを取り扱う機材もしくはサービスもまた高額であり，個人にとっては敷居が高い．

本稿は，このような状況にありながらも，個人によるツイートデータの収集・分析をまったく諦めるのではなく，まず制約はありながらもそれらを実現する手法を考案した．次にその手法の有効性を実証するために，日本国内で社会現象を起こし，大量のツイートを発生させた2つの話題に関するツイートの収集・分析を試みたので，その結果の概要を報告し，本研究が提案する手法がいわゆるビッグデータと呼ばれる研究の重要な一翼を担うことを示す．

2. 背景

2010年台の今，Twitterは，人々のほとんど生の（誰にも内容を編集されていない）所感や告知を，速報性のある状態で知ることができ，かつ世界規模で普及している有数の媒体であると言える．このような特徴もあり，報道媒体等の公共性のある組織もTwitterを世論分析の情報源として活用しているが，Twitterの情報源としての有用性は，このような大規模な組織に限ったものではない．たとえば自社商品のマーケティングに活用したいと考える中小企業や，世論の興味深い傾向を明らかにしたいと思っている個人やそれに類する小規模な研究チーム（以降，個人レベルと称す）にとっても有用である．

そこで筆者らはまず，ツイートデータ収集・分析に関する先行事例を調査した．実際にツイートの収集・分析を大規模に行っている研究者の話を聞いたり，同分野の関連論文（[1][2][3]など多数）を調べた結果，それらに共通しているのは，収集にTwitter Streaming API（Streamed Tweets）[4]を用いているという点であった．これは，世界中からTwitter上に投稿されるツイートのうち，そのすべて，あるいは一部をリアルタイムに取得できるものである．ただし一般ユーザに開放されているものは一部が取得できるもの（sample，filter等[4]）である上，収集対象となるツイートの発生頻度がAPIの基準を超える場合には間引かれてしまうため，大量に取得する必要がある場合には実用的でない．したがってこれら研究グループは，Firehose等，取得数の制限を受けないAPIを契約し，長期間に渡る大規模収集を行っていると考えられるが，個人レベルでその契約を結ぶことはほぼ不可能で，非現実的な選択肢である．

引き続き，個人レベルでの大量のツイートを収集する方法を調査した結果，東京大学の鳥海の2015年の報告書[5]を見つけた．この報告書では，ツイートデータ収集・分析に関する現状やデータ規模，収集・分析ツールの紹介など，これからTwitterデータ分析を始めようとしている者が把握すべき基礎知識の比較的新しい状況がまとめてられていたが，先程述べたStreaming APIを前提としない収集方法についても記されていた．

2.1 無料APIによる収集の制約

その報告書は，収集・分析ツールとしてTTC/TTM[6]，およびTTCの機能を拡張したWTCについて言及していた．特にTTC/TTMは，100を超える研究に利用されているとツールの作成者が述べている．

しかし，これらのツールはTwitter社が無料で公開している“Standard search API[7]”を利用しており，第一に，過去7日分までしか検索できないという制約がある．他にも，そのAPIに課せられたアクセス頻度制限（レートリミット）への対応方法の問題により，1日あたり約18,000ツイートを超える規模の話題を集めることが難しい．先程の報告書においても，古いツイートや大量のツイートに関しては有料で収集することが推奨されていた．

2.2 有料APIの費用問題

そこで，ツイートデータを有料で収集する方法を調査したところ，小規模な組織や個人にとっては非常に高額な費用を負担しなければならないことが分かった．

Twitter社と戦略的ソリューションパートナー契約を結び，日本国内におけるツイートデータ販売の代理店業務を行っているNTTデータが提供するAPI[8]の場合，過去約1年分のツイートデータを任意のキーワードで検索して配信する「ヒストリカルサーチAPI」という比較的安価なサ

ービスであっても，月額45万円以上，最低契約期間6カ月，初期費用別途とされており約300万円以上の費用負担が見込まれる．

また，本研究開始後ではあるが，2017年11月にTwitter社は，このような高額な費用負担を緩和するために“Premium APIs”というサービスを発表した．一例を挙げると，Twitterサービス開始当時まで遡れ，毎月125万ツイートまで取得できるプランでは月額1,899米ドル，遡れる期間が過去30日という制限を付ける場合は699米ドルである[9]．毎月取得可能なツイート数に応じて料金は149米ドルから用意されており，中・小規模のツイートを収集する場合の敷居は低くなった．しかし，社会現象と呼ばれるほど大規模な話題を分析するために大量のツイートを集めたいというニーズに応えるものにはなっていない．

なお，2018年8月に無料のTwitter API の使用制限の告知・実施があったが，本研究に深刻な影響を与えるものではなかった．1つはUser Streams APIが廃止された点であるが，これは自分のアカウントにリツイートや「いいね」が付けられた場合，フォローがなされた場合等に自動的に通知を受ける機能であって，ツイート検索機能を利用する本研究には無関係である．もう1つは，Twitter Appsというアプリケーション登録のサイトが廃止され，そこで提供していた機能がTwitter Devepolersに移行するという点である．移行に際し，申請するアプリケーションの概要や使用目的等を英語で明記して審査を受けるという手順が追加された．本研究では，4.1節で述べるアプリケーション認証方式を利用するためにアプリケーション登録が必要であるが，ツイートを収集・分析する研究目的で使用する旨の申請をして受理されることも確認できた．

2.3 本研究の目的

無料で使えるStandard search APIには主に2つの欠点がある．

- 過去7日前までのツイートしか収集できない
- レートリミットが厳しいために大量ツイートの収集には適していない

本研究では，Standard search APIを利用しつつ（すなわち個人には現実的ではない費用を発生させず），後者の欠点を克服することを目標とした．そのために，実際に社会現象と呼ばれる話題で生じるツイートデータ規模の確認，ツイートデータを収集・分析する手法の提案を行った上で，社会現象とされた実際のテーマにもとづくツイートを収集・分析し，提案手法の有効性を検証した．

3. 「社会現象級」ツイート収集の実現可能性

3.1 「社会現象級」ツイートの規模

先に記したとおり，NTTデータは日本におけるツイートデータの販売業務を行っているため，現在までのツイートデータも自由に参照できる立場にあるものと思われる．実際に同社は「イマツイ」というブログサイトを運営し，全ツイートデータを参照して得られた興味深い分析結果を公表している．本稿では，特に断りのない限り，全ツイートデータを参照できる立場にあるNTTデータが前述のブログサイトで取り上げた用語（流行語）を「社会現象級」の「話題」と呼ぶことにする．ただし，このブログは日本語圏の読者に向けたものであるので，日本語圏における社会現象かつツイート傾向であることに気を付けなければならない．

イマツイにて2016年12月28日に公開された記事[10]によれば，2016年の元日から同年12月12日までに発生した全ツイートを調べ，音楽，映画，流行語，テレビドラマという各ジャンルで話題になった語が含まれるツイート数を数えたところ，1位が「ポケモンGO」であり，これを含むツイート数が1929万8506（リツイートは除く），同様に2位が「PPAP」であり，1091万2400であった．どちらも同年に世界的に流行し，社会現象と呼ばれるほど話題になったコンテンツであったが，それら社会現象級のツイート数は年間1000万のオーダであることが分かる．

ただし年間を通してツイート頻度が一定であったとは考えづらく，実際，同記事によれば1位の「ポケモンGO」は同年7月が最も多く，その数は912万6484と報告されている．これを1日あたりとして平均すればおよそ30万ツイートになるが，もちろん日によって差はあるはずである．仮に最も多い日と少ない日の差が10倍に達していたと仮定すると，この話題で1日で300万ツイート（リツイート除く）が発生した日もあったことになる．

3.2 無料APIの収集速度

次に，無料で使えるStandard search APIの収集速度を検証する．

Twitter社が公開している仕様[7]によれば，このAPIの1回呼び出しで最大100ツイート収集できる．そして，レートリミットによって，15分あたり450回まで呼び出せる（アプリケーション認証の場合）．1秒あたりに換算すれば，毎秒50ツイートまで取得できる．これは，1日当たりに換算すれば432万ツイート，1か月あたりでは1.3億ツイート，1年あたりなら約15.8億ツイートまで取得できる計算になる．よって単純計算では，前述の社会現象級のツイートの発生量と比較すると，1つの話題に絞りさえすれば十分収集可能といえる．

もちろん，先程の「ポケモンGO」ツイート数の見積もりで仮定した多い日と少ない日のツイート数の差が実際には10倍以上である可能性や，リツイートを含めるとさらに数が増える可能性，さらにAPI仕様書どおりの性能が長期間は持続しないなどの可能性を考慮すると，実現できない可能性もある．しかし，このように厳しく見積もれば1日あたりの収集限界ツイート数を超える可能性があるものの，厳しい条件が何日も持続するとは考えにくく，実際は収集できるのではないかと予想した．

4. 提案する大量ツイート収集手法

4.1 収集時点でのツイートデータ絞り込み

先程の社会現象級ツイートに関する考察により，1つの話題に絞れば無料のAPIで集めきれる可能性があることを確認した．そこで，全世界で発生するツイートをとりあえずすべて集めるという方針ではなく，Standard search APIを利用してツイートの取得段階で必要なツイートに絞り込むことを基本方針とする．これは，2.1節で引用した報告書に記されていた方針と同じである．

ただし認証方式は，ユーザ認証ではなく，アプリケーション認証を用いるという違いがある．ユーザ認証によってStandard search API を利用した場合，そのユーザがフォローしている鍵付きアカウントの限定公開ツイートも収集でき，逆にブロックされているアカウントのツイートは収集できない等，ユーザの閲覧権限に応じて収集対象となるツイートの範囲が変化するが，大量収集の場合においてはそのような限定公開ツイートは不要である上に，レートリミットが厳しく設定されている（毎秒に換算すると最大20ツイート）．アプリケーション認証にすると，ユー

ザ認証ではないため，未ログイン状態で閲覧可能な範囲のツイートが取得され，かつレートリミットも毎秒50ツイート相当に緩和される．その結果，3.2節のとおりの収集速度が引き出される．

4.2　ツイートID管理を厳密にした降順収集

ツイートの収集は，**図1**のようにして行う．

図1　Standard search APIによるツイート収集の基本方針

Standard search APIでは，指定された日付（UTCにおける午前0時0分0秒）またはツイートIDを起点としてツイートID番号の降順（過去に遡る方向）にツイートを収集できる．そこで，n+1回目の呼び出しでは，n回目の呼び出しで収集されたツイートの中から最も小さいID番号から1を引いたものを起点にして呼び出せば連続的に集められる．ただし，レートリミットを超えないようにするため，呼び出し間隔が2秒になるように，呼び出し時間を調整しながら呼び出す．

これを繰り返せば，理論上最大7日前までのツイートが集められる．ただし，収集作業をしている間に7日前より古くなってしまうツイートもあり得るため，初回は7日前の日付を起点として遡れるだけツイートを収集し，それ以上遡れなくなったところで次は6日前の日付を起点にして，すでに取得済のツイートID番号に達するまで行う．長期間に渡って収集を行う場合には，収集を行った日の分まで集め終わった後，翌日のUTC午前0時0分0秒を過ぎるのを待ってから翌日の日付を起点にして同様の収集を行えばよい．

ツイートを過去に遡りながら収集するという基本方針も，2.1節で引用した報告書で述べられていたツールが採用していたものと同じである．しかし，一定量以上のツイートを収集するにはユーザがその都度収集実行のための収集ボタンを押さなければならない上に，レートリミットを超えないようにボタンを押すタイミングをユーザに任せていること，そして収集ボタンを使って集めた後にツイートIDが保存されないため，続きのツイートから集められないといった問題があり，大量収集に適した作りになっていなかった．本研究では，4.4節で述べるようにツイートIDを保存して，それらの問題を解消したツイート収集プログラムを作成した．

4.3　持続性の高い収集プログラムおよびツイートデータフォーマットの実現

Standard search APIでは過去7日前のツイートまでしか遡れないため，たとえば1年分のツイートを収集するには，ほぼ1年間収集作業を行わなければならない．長期間に渡って同じプログラムを使う場合に注意しなければならない問題の1つに，プログラムやデータの互換性や持続性がある．

もし収集作業期間中のある日に，たとえば収集プログラムを動かしているOSをアップデートしたことが原因と疑われる理由で動かなくなったら，収集作業ができなくなってしまう．プログラムが使えなくなってしまうような環境変化は，他にもさまざまな要因によって起こり得る．したがって，長期間に渡る収集を行う際には特にプログラムあるいは実行環境の持続性・互換性に注意を払うべきである．

また，持続性や互換性という性質は，プログラムのみならず収集されたデータにとっても重要である．私達が収集した貴重なデータを，数十年後に後世の研究者が，当時の世論を研究する貴重な情報源として利用する可能性もある．その時，データフォーマットがすでに廃れたものになっていたら，彼らはデータを開くのに苦労したり，最悪諦めねばならないかもしれない．

以上の懸念から，収集プログラムを作るにあたり，筆者らは「POSIX中心主義」を採り入れた．

4.3.1　POSIX中心主義

POSIX中心主義[11]とは，互換性や持続性に最大限の配慮をしてプログラムやデータを環境変化に強くするためのプログラミング指針であり，2017年，我々は当該論文にて実際に互換性・持続性向上に効果が得られたことを報告した．

基本的にはPOSIX文書に記されている範囲でプログラミングすることにより，多くのOSで動作する互換性，そしてPOSIXの改訂が非常に少ないゆえの持続性を獲得する．また，POSIXの範囲で実装できない部分に関しては，POSIX範囲外のソフトウェアへの依存も認めるが，同一機能を持った複数のソフトウェアに対応させることを必須とし，RAIDや電源二重化のような冗長性を持たせ，結果として高い持続性を確保できる．

この考え方に基づくなら，取得したツイートの保存先としても，Oracle，MySQL等のRDBMSと呼ばれるミドルウェアを使うことは避け，POSIXの範囲で規定されているファイルやディレクトリの枠組みと，POSIXの範囲のUNIXコマンドのみで構成することが望ましい．そこで，次のようなデータ格納方法を考案した．

4.3.2　RDBMSミドルウェア非依存のデータ管理

まずは，取得した各ツイート1つ1つの格納方法に関する議論である．Standard search APIからは仕様書[7]に記されたとおりのJSON形式（ツリー構造データ）で得られるが，多くのUNIXコマンドは，行と列からなる二次元データの処理に向いた仕様になっており，そのままでは扱いづらい．そこで，取得する段階で主要な情報（ツイートID，投稿者，本文等）を行や列に再配置して格納する．本研究では次の2つの形式を用いることとした．

1つ目は，「RESデータ」と名付けた形式である．

これは，**図2**のように1つのツイートが日時の行から7行で構成され，各行の構成は次のとおりである．

図2　ツイート格納形式（RESデータ）の例

第1行　ツイートされた日時（タイムゾーンは収集したホストのもの）
第2行　ユーザ名とユーザID（スクリーンネーム）．認証済みアカウントの場合は行末に“[v]”が付く．
第3行　ツイート本文．公式リツイートの場合は1文字先頭に“RT”が付き，さらに1文字字下げされる．
第4行　収集時点でのリツイート数といいね数．自分がリツイートやいいねをすでにしていれば“RET”や“FAV”と表示される．
第5行　場所の名前と緯度経度．多くのツイートにはこの情報がなく，その場合は“-”と表示される．
第6行　Twitterクライアント名とその公式ページURL
第7行　そのツイートのURL

RESデータは，人がmoreコマンド等で直接閲覧するのに向いた形式である．

2つ目は，「ANLデータ」と名付けた形式である．こちらは1ツイートの情報（構成要素）を次のように1行に並べたものである．

第01列　ツイートされた日時（“YYYY/MM/DD-hh:mm:ss”，タイムゾーンは収集したホストのもの）
第02列*　ユーザ名
第03列　ユーザID（スクリーンネーム）
第04列　認証済アカウントなら“v”，それ以外は“-”
第05列　公式リツイートなら“RT”，それ以外は“-”
第06列*　ツイート本文
第07列　収集時点でのリツイート数
第08列　収集時点でのいいね数
第09列*　ツイート場所の名前（ない場合は“-”）
第10列　緯度，経度（ない場合は“-”）
第11列*　Twitterクライアント名
第12列　Twitterクライアントの公式ページURL
第13列　そのツイートのURL

1行1ツイートであるので，ツイート数を数えるにはテキストデータの行数を数えればよく，また特定の構成要素だけ抽出したければAWKコマンドで列番号を指定するだけで簡単に行える．上記の列番号に「*」が付いているものは，文字列中の空白等がエスケープ（空白は“_”に，タブは“\t”に，“_”や改行文字はバックスラッシュ付文字にエスケープ，バックスラッシュは二重化“\\”）されて格納されていることを示す．これにより，AWK等での列番号指定が狂うことを避けている．また，AWKを**図3**のように書けばエスケープを解くことも簡単である．

```
awk '{ # エスケープ解除の際,文字同士の衝突を避けるため,
       # ツイートには絶対に含まれない制御文字を一時利用
       s=$6;
       gsub ( /\\\\ /  , "\001" , s ) ; gsub ( /\\ _ /     , "\002" , s ) ;
       gsub ( /\\t /   , "\003" , s ) ; gsub ( /\\n/       , "\004"  , s ) ;
       gsub ( / _/     , "  "       , s ) ;
       gsub ( /\004/ , "\n"     , s ) ; gsub ( /\003/     , "\t"     , s ) ;
       gsub ( /\002/ , "_"      , s ) ; gsub ( /\001/     , "\\"     , s ) ;
       print       s ;                                                  }'
```

図3　ANLデータからツイート本文を抽出する例

なお，ANLデータは，機械（シェルスクリプト）がデータを加工・分析しやすいように考案した形式である．次に，それらのツイートデータをどのように整頓するかを議論する必要がある．大量のツイートを取り扱うことを想定する場合，すべてのツイートデータを1つのファイルに格納するのは，閲覧するにも編集するにも非効率であるが，1ツイート1ファイルで管理するのもまたファイルオープンの処理コストを考えれば非効率であり，適切な粒度に分割して管理する必要がある．

そこで筆者らは，1つの年月日時分秒で1ファイルとし，年月日時分秒という時系列にデータを整頓する方式を考案した（**図4**）．

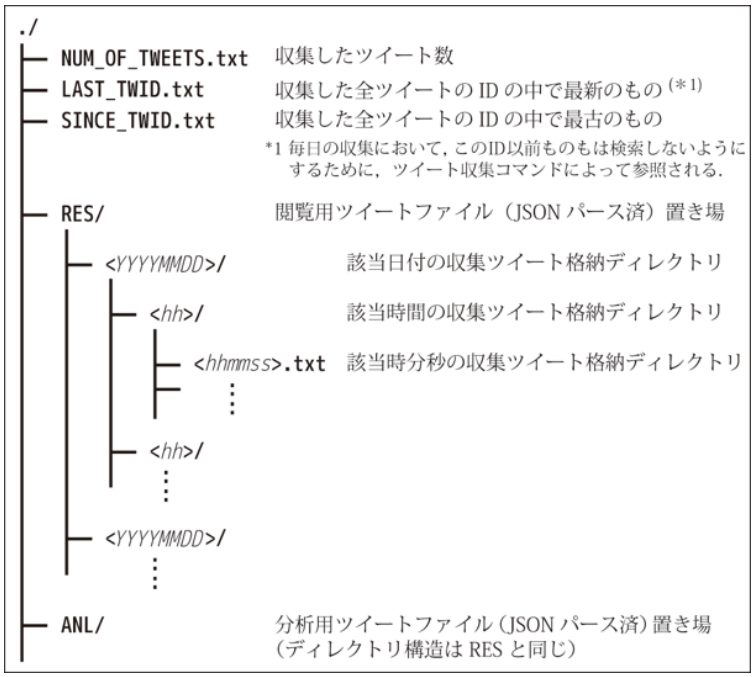

図4　大量ツイート格納のためのディレクトリ構成

年月日ディレクトリの下に時ディレクトリを作り，その中に時分秒のファイル名を作る．このようにしておけば秒の粒度で簡単にツイートを抽出できる一方で，どんなにツイートが大量に投稿されても1日あたり最大で86,400（1日の秒数）個しかファイルは増加しない．年月日ディレクトリの下に直接時分秒ファイルを置かず，時ディレクトリを挟んだ理由は，多くのUNIX系OS上で1つのディレクトリ直下に置くファイル数が1万個程度を超えると急激にファイルアクセスの効率が悪化するという経験に基づいている．なお，このような時系列で整頓するディレクトリを，先程のRESデータ，ANLデータそれぞれに作成した．

このような時系列に整頓されたディレクトリにはまた，別の利便性もある．何らかの話題のツイートを分析する時，いつからいつまでの期間を対象とするのかは必ず決めなければならないため，指定された開始日時から終了日時までのツイートデータを簡単に切り出せるという点でも，本格納形式は便利であった．

4.4　大量ツイート収集プログラムの作成

以上の方針に基づき，収集プログラム「小鳥男」[12]を作成した．POSIX中心主義に基づいているため，プログラムの大部分はPOSIX版sh互換のシェルスクリプトである．このプログラムは目論見どおりに多くのUNIX系OS上で動作することが確認された．また，シェルからコマンドとして実行できるため，日々の収集作業の自動化も容易に行えた．

そして，4.3.2項で説明した方針に基づき，ツイートの投稿年月日および時分秒に応じてディレクトリとファイルを分け，日時に基づく検索性を高めるとともに，POSIX規格で用意されている基本的なUNIXコマンドで簡単に取り扱えるよう，半角空白区切りのプレーンテキスト形式を基本とした[13]．

5. 社会現象級ツイート収集・分析の検証

前節の提案手法に基づき，社会現象と呼ばれた2つの話題に関するツイートの収集・分析を実際に行った．

5.1　「バルス」ツイート

その1つは，アニメ映画「天空の城ラピュタ」[14]がテレビで放送される度に大量投稿される「バルス」という語を含むツイートの収集による検証である．

「バルス」とは，劇中で登場する呪文であり，主人公らがこの呪文を唱えるシーンが放送される瞬間に合わせ，多くの視聴者がこの呪文を含むツイートを一斉に投稿することが恒例になっている．過去にその影響でTwitterのサービスがダウンしたこともあり，ダウンを狙って投稿する者も増え，2013年8月2日の再放送時には，その瞬間に秒間ツイート投稿数（バルス以外も含む）が14万3199を記録した[15]．

このように，バルスは瞬間的に大量のツイートを発生させるイベントであり，これが収集できるかどうかも本提案手法の実用性を検証するうえで重要な指標の1つになる．

5.1.1　ツイートデータ収集

まず，対象としたバルスイベントは，2017年9月29日放送分の天空の城ラピュタである．

そして，この検証で使用したコンピュータの主なスペックは次のとおりである．

- CPU：AMD Ryzen 1700
- メモリ：16GB
- HDD：1TB
- OS：WSL（Windows 10上で動くLinux環境）
- インターネット回線：NTT東フレッツ光（100Mbps）

これは，個人向けに販売されている一般的なPCおよびインターネットプロバイダで構成された環境である．

また，ツイート収集に用いた検索クエリ（前述の小鳥男を介してTwitter APIに送るもの）は次のとおりである．

> バルス OR "バルス祭り" OR #バルス OR
> #バルス祭り OR ヴァルス OR #ヴァルス OR
> barusu OR #barusu OR bals

このようないくつかのパターンで表記揺れが起こるものと想定し，バルスを意図していると思われるものは一通り収集されるようにした．

5.1.2　分析

ここでの分析は，収集されたツイートの毎分の数を求めることとした．NTTデータ「イマツイ」が天空の城ラピュタ再放送の都度，同様のデータを公開しており，両者の収量を比較できるためである．

毎分のツイート数を数えるにあたっては，格納されているツイートデータが4.3.2項で述べたように，UNIXコマンドで扱いやすい構造になっているため，**図5**のような簡単なシェルスクリプトで可能である．

```
cd /PATH/TO/TWEET_DATA/DIR
find . -name '*.txt'          |
xargs cat                     |
awk '{print substr($1,1,16)}' |
sort                          |
uniq -c > count_TPM.txt
```

図5　1分ごとのツイート数を数えるシェルスクリプト

ちょうど年月日時分までディレクトリ分けされており，その中にあるテキストファイルが1ツイート1行になっているため，各ディレクトリの行数を数えている．

その結果，番組放送中に観測されたバルスを含むツイートの1分あたりの数は，**図6**のように推移していたことがわかった．なお，本研究で収集された数と，NTTデータが公開している数，そして両者の差を併記してある．

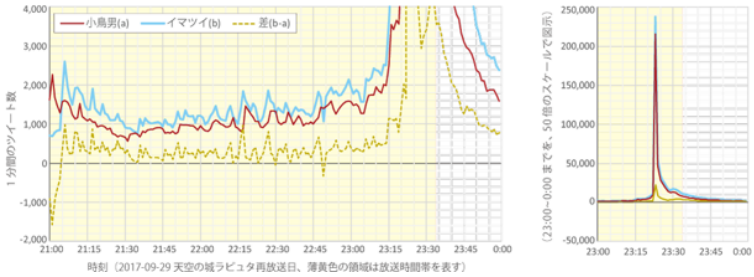

図6　2017-09-29 天空の城ラピュタ(再放送)で観測された「バルス」等を含むツイート数

当日の「バルス」ツイート大量発生のピークは23時23分であったが，この1分間の収量は，小鳥男が21万5245個，NTTデータが23万7295個で両者の差は10%未満であった．また，放送中の全収集ツイート数は同様に，59万2382，70万4893で約16%の差であった．両者の差の原因は，主に検索クエリによるものではないかと考えられるが，NTTデータ側のクエリは公開されていないため断定はできない．

5.2　「けものフレンズ」ツイート

もう1つは，アニメ「けものフレンズ」やそれに関連した主要なキーワードを含むツイートの収集による検証である．

「けものフレンズ」[16]は，2017年の1～3月にテレビ放送されて社会現象と呼ばれるまでに人気を博したアニメ作品であり，Twitterトレンド大賞2017のアニメ部門で大賞にも選ばれた．この賞は，Twitterトレンド大賞実行委員会がTwitter JAPAN協力のもとに，日本国内4500万人のTwitterユーザが実際に投稿したツイートを分析することで映画・ゲーム・アニメ・ニュース等の部門別に最も流行した語を見つけ出し，授与したものである．公式ページには「4500万人がつくった忖度なしの今年のトレンド」という説明があることから，授賞対象は人為的に選んだのではなく，機械的な分析結果に基づいて選定されたものと思われる．

さらに，NTTデータ「イマツイ」によれば，けものフレンズに関するツイートは，2017年に新語・流行語大賞にノミネートされた言葉の中で最も多いツイート数を発生させた語で，その数は年間約5000万件であった[17]．

前節のバルスが瞬間的なコンテンツであったのに対し，けものフレンズは年間を通して話題が持続したコンテンツであり，かつツイート数に関する賞を受賞したものである．本提案手法によるこのコンテンツの収集・分析の実用性が示せれば，大多数のコンテンツ（少なくとも日本語圏）に通用すると言えるため，その意義は大きい．

5.2.1　ツイートデータ収集

主要なけものフレンズ関連語（後述）を含むツイートの収集は，2017年3月20日から2018年3月19日までの1年間実施した．この開始日は，アニメ「けものフレンズ」の全12話中第10話まで放送された後であるが，本提案手法の検証に利用しようと起草したのがこの頃であったため，第1話からの収集には間に合わなかった．

また，Twitterトレンド大賞2017を受賞したことや，年間約5000万ツイートを発生させたという事実も，収集開始後に知ったことである．本提案手法は，7日前より古いツイートの収集ができないという制約を解決するものではないので，ツイート数に関する賞を獲得すること（本提案手法の実用性を示せること）が分かっていて収集し始めたわけではないことに注意されたい．

そして，この検証で使用したコンピュータの主なスペックは次のとおりである．

- 2017年3月から12月
  — CPU：Intel Core 2 Duo E8400
  — メモリ：4GB
  — HDD：1TB
  — OS：CentOS 5.11
  — インターネット回線：NTT東Bフレッツ（100Mbps）
- 2018年1月から
  — CPU：Intel Core i5-6500
  — メモリ：4GB
  — HDD：3TB
  — OS：FreeBSD 11.1
  — インターネット回線：NTT東Bフレッツ（100Mbps）

長期間収集作業を行う事情により，バルスの検証時とはまた別のコンピュータを使用した．また，OSが古くなっていたため，途中から別のコンピュータにデーター式を移して収集・分析作業を継続させた．

ツイート収集の際に使用した検索クエリは次のとおりである．

> けものフレンズ OR けもフレ OR けもふれ OR
> ケモフレ OR （kemono friends） OR
> じゃぱりぱーく OR ジャパリパーク OR
> "サーバルちゃん" OR "かばんちゃん" OR
> "ぺパプ"

時が経つにつれて関連語は増えていくが，今回の検証では上記のクエリで固定した．

5.2.2 分析

バルスツイート分析の際に行った毎分のツイート数を数える分析も含め，主に次のような分析を行った．

- 毎分のツイート数の推移をグラフ化し，極大点がどの日時に生じたか，およびどんな話題が原因で極大になったのかを調査
- 同様のことを毎秒のツイート数で実施（主に30分アニメ放送中に，どのシーンで極大が生じたかを秒単位で特定する場合）
- 1週間分の全ツイート本文を形態素解析して頻出語のランキングを作成し，その週で話題だったキーワードを調査
- 同様のことを番組放送時間帯で実施（番組中に最も話題になった登場人物の調査等のため）
- 一部のツイートに含まれる緯度・経度情報を地図にマッピングし，多数ツイートが投稿された場所を調査

すでに述べたとおり，ツイートデータはUNIXコマンドで扱い易いプレーンテキスト形式で格納されているため，これらの分析でもその大半は，POSIX規格で用意されているUNIXコマンドや簡単なシェルスクリプトで実現できた．

たとえば，ツイート数推移グラフのある極大点について，どんな話題がその原因になっていたのかを調べるには，**図7**のようなコマンドを実行すればよい．グラフを作図して明らかになった極大日時“YYYY-MM-DD hh:mm”に該当するファイルを一通り開き，AWKコマンドでツイート本文の文字列を取り出し，あとはsortとuniq -cの組み合わせで同一文字列の出現回数の多いものを列挙すれば，どんなツイートがその瞬間にリツイートされて極大を生じさせたかが分かる．一方，テレビで盛り上がるシーンが放送されるなどして，リツイートではないツイートがたくさん投稿されて極大が生じる場合には，単純にlessコマンド等で該当日時のツイートを閲覧すれば大抵は理由が分かる．

```
cd /PATH/TO/TWEET_DATA/DIR
cat YYYYMMDD/hh/hhmm* |
awk '{print $6;}'      |
sort                   |
uniq -c                |
sort -k 1nr,1          |
head
```

図7　極大点の原因になったリツイート文を調べるコマンド

POSIXコマンドではできない作業は，グラフやツイート発生場所マップ（後述）の作成と形態素解析であった．グラフ作成には表計算ソフト（今回はMicrosoft Excelを用いた）を，地図作成には無料のWebマッピングサービス，形態素解析にはMeCabコマンドを用いた．ただし頻出単語の数を数える場合には，MeCabコマンドで単語に分解した後，やはり先程と同様にsortとuniq -cの組合せで行える．

図8は，このようにして作成された実際の分析例で，2017年9月18～24日の1週間分における「けものフレンズ」関連ツイートを分析したものである．

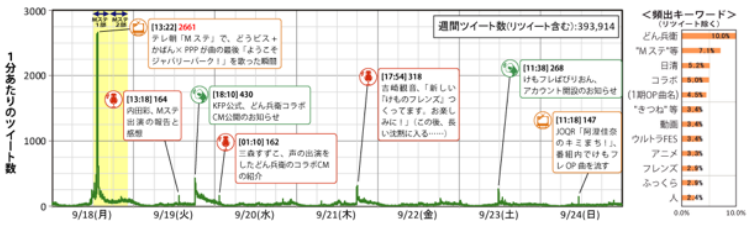

図8　2017-09-18週の，毎分ツイート数推移（左）と，全ツイート中のキーワード出現割合（右）

これは2つのグラフから構成されており，左が分間ツイート数の推移，右が頻出語を出現頻度（登場した延べ回数を全ツイート数で割ったもの）順に並べた棒グラフである．

左の折れ線グラフでは，9月18日に顕著な極大点が出ており，この時刻のツイートを調べた結果，音楽番組のミュージックステーション（Mステ）にけものフレンズの出演声優が登場して歌唱している時間帯だった．けものフレンズの人気が本放送終了半年後でも依然高いことやテレビが瞬間的な話題を起こしやすい性質のメディアであることを示唆する結果が得られた．

ところが右の棒グラフを見ると，この1週間の間，「Mステ」（「ミュージックステーション」等の同義語と合算）よりも「どん兵衛」というキーワードの方が話題にされた機会が多かったことが分かる．これは翌19日にインターネット上でけものフレンズと日清どん兵衛がコラボしたCM動画作品が発表された影響であった．この動画はいつでも視聴可能な状態で公開されたが，インターネットがテレビとは対照的に，瞬間的ではないが継続的に話題が広まりやすい性質のメディアである様子が読み取れる．

このほかにも，位置情報付のけものフレンズ関連ツイートの緯度・経度をマッピングしたところ，各地の動物園から投稿されているものが多いなど，興味深い結果がいろいろ明らかになった[18][19]．

6. 提案手法によるツイート収集・分析の実用性

今回実施した，実際の社会現象ツイートの収集・分析検証実験から，提案手法がどの程度有効であるか確認することができた．

6.1 バルスツイート収集からの考察

最初に1つの重要な知見が得られた．それは，バルスのように瞬間的に大量のツイートを発生させるイベントであっても，100万ツイートのオーダに収まるということである．Standard search APIでは1秒あたり50ツイート収集可能であるため，理論上，5〜6時間程度で収集できることになるが，実際にも，今回の59万2382ツイートの収集に要した時間は3時間26分であり，予想を裏付ける結果となった．この所要時間は，APIに課せられた過去7日間という制限期間を遥かに下回っている．

また，59万2382ツイートを3時間26分で収集できたという結果から，ほぼAPIのレートリミットに近い頻度でツイートが収集できていたことが確認され，用意した通信回線の速度に問題がなかったことも分かった．実際に，バルスツイートの収集時にどれだけの通信速度を必要としていたかを計算してみると，APIがこちらに送出したJSONの生データは全部で約3.0GBあったので，これを3時間26分で除すれば2.0Mbpsということになる．元々APIのレートリミットによる制約が大きいため，現在のところ，通信回線は実測値が数Mbps以上であれば滞りなくデータを収集できるということである．

このように，瞬間的に大量のツイートを発生するイベントに対して以下に記す点を示せたことから，瞬間的に大量のツイートを発生させるイベントに対する本提案手法の有効性が実証できた．

- Standard search APIが課す制限期間に対し，十分早く収集できること
- 収集されたツイート数は，信憑性の高いデータと突き合わせても16%の差に収まったこと

6.2　けものフレンズツイート収集からの考察

一方，けものフレンズ関連ツイートによる検証実験からも重要な知見が得られた．NTTデータ「イマツイ」は，けものフレンズの2017年のツイート総数は約5000万と発表していたものに対し，本検証実験により収集されたツイート数は2316万6588であり，半分程度だった．しかし，本研究では3月下旬からの一年間であって収集次期が若干遅く，また検索クエリが異っていたこと（イマツイのクエリは非公表，特に本研究では検索クエリにハッシュタグを含めていなかった）ことを考慮すれば，想定される差であり，収集時期と検索クエリを揃えられればイマツイの結果にほぼ近づけられた可能性がある．

次に，本提案手法により収集し，格納されたツイートデータの総サイズは約89GBとなり，社会現象級のツイートであっても2018年現在の個人向けパソコンで十分取り扱える規模に収まっていることが確認された．ただし，Twitter APIから送られるJSON形式の生データ（本研究で利用したRES/ANL形式に変換する前のデータ）は2倍程度のサイズであったため，こちらも保存する場合には前述の2倍程度の領域を見積もる必要があることを補足しておく．

また，1年間毎日収集を実施してきたが，けものフレンズにおいては1日分を収集するのに1日以上要することは一度も起こらなかった．ツイート数が最も多かったのは2017年3月29日（最終話放送日）であったが，この1日分（日本時間の2017年3月28日午前9時から2017年3月29日午前9時）で収集されたツイート数は77万6796であり，4時間19分で集められた．

長期間に渡る収集となれば，OS上のソフトウェアアップデート等のホストのメンテナンスが避けられないが，毎日の収集時間にこれだけの余裕があればそれらの作業時間を確保することもできる．実際に，2018年1月に収集するコンピュータを更新したが，更新作業の時間を問題なく確保でき，そしてホストやOSが更新された後もプログラムはまったく問題なく動作し，収集・分析作業に支障を来たすことがなかったことも重要な点である．仮に，1日分の収集に1日以上かかることあっても，それが連日続くようなものでなければ，前述のようなメンテナンス作業は毎日起こるようなものではないため，余裕のある日に実施すればよい．

このように，継続的に大量のツイートを発生するイベントに対しても以下に記す点が示せたことから，継続的に大量のツイートを発生させるイベントに対しても本提案手法が有効性であることが実証できた．

- 信憑性の高いデータと比較した結果，ほぼ同量のツイート数を収集できたこと
- 収集されたツイートデータのサイズは，検証を行った2017年現在，すでに個人で容易に扱えるサイズに収まったこと
- 1日分を収集するのに1日以上要する事態は一度も発生しなかったこと
- 収集作業期間中に，コンピュータの更新等のメンテナンス作業時間を確保できたこと
- コンピュータを更新しても，プログラムは正常動作し，作業に支障が出なかったこと

また，この提案手法の基本方針は，ビッグデータ分野の研究を遂行する上でも応用可能である．分析に必要なデータのみを早い段階で選択・抽出することはもちろん，プログラムやデータフォーマットを平易な構造にし，分析データおよび分析作業環境の持続可能性を高めることは，同分野においても求められると考えられ，その際には本研究で得られた知見が活かせる．

6.3　令和ツイート等，その後の収集事例と計画

2019年4月1日に新元号が「令和」であると発表されてからすぐに「"令和" OR "れいわ" OR "レイワ" OR "reiwa" OR #令和 OR #れいわ OR #レイワ OR #reiwa」を検索クエリとして令和ツイートの大量収集を開始した（収集には2018年1月以降のけものフレンズツイート収集環境を利用）．新元号への改元というイベントは，前出のものよりもより多くの人々に知られ，ツイート規模（ツイート総数や発生頻度，話題の持続期間）も大きなものになると予想され，本提案手法が引き続き通用するかを検証する必要があると考えたからである．

発生数と頻度が特に高かったのは4月1日11時41分の発表直後と，5月1日0時の改元の瞬間であった．後者の方がより多く，収集できたツイート数は5月1日0時0分2秒の1秒間に1万4539ツイート，それを含む0時0分の1分間に28万7961ツイートと，5月末時点までの最高頻度を記録した．その日一日は全体的にもツイート数が多く，本研究によるツイート収集をしてきた中で初めて1日分のツイート収集に1日以上を要した．しかしその後は発生頻度が次第に落ち着いていき，5月3日には収集対象のツイート発生日時が収集日時に追いついた．発表された4月1日から5月末までの2か月間に収集されたツイート総数は2934万2405である．以上が概要であり，当該ツイートについては引き続き詳しい収集・分析を行っている．

「令和」という日本中で話題になるイベントであっても，本手法による大量ツイート収集はなお有効であり，7日前までというAPI取得制限に照らし合わせてもまだ十分に余裕がある．また，2020年には東京オリンピックという国際イベントが開催されるため，今後この話題についても検証し，本提案手法が日本語圏のみならず国際的な話題にも有効であるかどうかを検証する予定である．

7. まとめ

ツイートの分析は，多くの人々が，多くの視点から分析するほど，社会にとって興味深い結果がより多く明らかになる．よって誰でも簡単に始められるように裾野を広げるべきと筆者らは考えている．しかしながら，社会現象等の膨大な量のツイートを発生させる話題を取り扱おうとした場合，費用的，技術的な問題から，それが可能なのは十分な予算や設備を持った大企業や組織に限られていた．そこで本研究では，個人にとっても実用的な大量ツイート収集・分析の手法を提案し，それにもとづく収集プログラムを作成することで，本手法がビッグデータと呼ばれる研究の重要な一翼を担うことを示せた．

また，本提案手法に，POSIX中心主義というプログラミング指針を採り入れた理由の1つは，データ分析の裾野を広げるためである．多くの環境で，簡単に，かつ長期間使える手法であれば，より多くの人々がツイート分析を始められるようになるであろう．筆者らは数多くの興味深いツイート分析が報告されることを期待している．

謝辞　本手法を支持し，本稿投稿にあたり御指導くださった金沢大学の共同研究者の皆様，USP研究所の皆様に，心より感謝を申し上げます．

参考文献

1）Mathioudakis, M. and Koudas, N. : Twitter Monitor : Trend Detection Over The Twitter Stream, ACM SIGMOD Conference 2010, pp.1155-1158 (2010).
2）Walther, M. and Kaisser, M. : Geo-spatial Event Detection in The Twitter Stream, Advances in In- formation Retrieval, pp.356-367 (2013).
3）関　堅吾，金子崇之，山下真一：OSSを活用したTwitterデータ提供システムの構築，ディジタルプラクティス5（2）, pp.110-119, (2014-04-15).

4）山本裕介：第3回Twitter API 勉強会―ストリーミング API #twtr hack （オンライン）（2012-01-26）, https://www.slideshare.net/yusukey/3twitter-api-api (参照 2019-06-03)
5）鳥海不二夫：Twitter上のビッグデータ収集と分析，組織学会，組織科学 48（4）, pp.47-59 (2016-04-07).
6）松村真宏：TTM : TinyTextMiner β version (オンライン), http://mtmr.jp/ttm/ (参照 2019-06-03)
7）Twitter,Inc. : Standard Search API (オンライン), https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets (参照 2019-06-03)
8）NTTデータ：Twitterデータ提供サービス サービスメニュー （オンライン）, https://nazuki-oto.com/main/ (参照 2019-06-03)
9）Twitter,Inc. : Pricing – Twitter Developers (オンライン), https://developer.twitter.com/en/pricing (参照 2019-06-03)
10）NTT データ イマツイ：全量データで振り返る，2016の激動！（オンライン）, http://imatsui.com/trend/201612/ (参照 2019-06-03)
11）松浦智之, 大野浩之, 當仲寛哲：ソフトウェアの高い互換性と長い持続性を目指す POSIX 中心主義プログラミング，デジタルプラクティス Vol.8, No.4, pp.352-360, (2017-10-15).
12）秘密結社シェルショッカー日本支部：恐怖！小鳥男（オンライン）, https://github.com/ShellShoccar-jpn/kotoriotoko (参照 2019-06-03)
13）秘密結社シェルショッカー日本支部：擬似リアルタイム Twitter 検索・収集コマンド“gathertw.sh”使い方（オンライン）, https://github.com/ShellShoccar-jpn/kotoriotoko/blob/master/APPS/gathertw.md (参照 2019-06-03)
14）宮崎　駿（監督）：天空の城ラピュタ（DVD）, JAN:4959241980144
15）Twitter Japan：2013-08-03 12:01:14（JST）のツイート（オンライン）, https://twitter.com/i/Web/status/363494742518013952 (参照 2019-06-03)
16）けものフレンズプロジェクト：けものフレンズ（オンライン）, http://kemono-friends.jp/ (参照 2019-06- 03)
17）NTT データ イマツイ：リアルな話題量ランキング『イマツイ ツイート大賞2017』を発表！（オンライン）, http://imatsui.com/seasonal_topics/post_143/ (参照 2019-06-03)
18）松浦リッチ研究所：ついったーちほーの最大瞬間風速（2017年8月11日発行）.
19）松浦リッチ研究所：フレンズかんそくたい（2017年12月29日発行）.

松浦 智之（正会員）richmikan@richlab.org
プログラマ，テクニカルエンジニア（ネットワーク）等の本業の傍ら，主にその分野の同人誌を作り，コミックマーケット等に出展する活動を展開中．2016年より大学コンソーシアム石川にて「シェルスクリプト言語論」を開講し，講師も務めている．

當仲 寛哲（正会員）koho@usp-lab.com
UNIX哲学に基づくグルーテクノロジー「ユニケージ開発手法」考案者．東京大学卒業後，（株）ダイエーで基幹システムを刷新し業務改革．2005年ユニケージ開発手法の研究・教育・普及を行うUSP研究所を設立．代表取締役．

大野 浩之（正会員）hohno@staff.kanazawa-u.ac.jp

金沢大学総合メディア基盤センター教授．博士（理学）．東京工業大学大学院講師，郵政省通信総合研究所（現NICT），内閣官房併任を経て2006年より現職．非常時情報通信，情報セキュリティ，ものづくり支援等の研究に従事．

投稿受付：2019年3月11日
採録決定：2019年8月2日
編集担当：宮下健輔（京都女子大学，IOT研究会主査）