

卒業論文 2010年度(平成22年)

マイクロブログを用いた
キーワードと地理的位置の対応付けシステム

慶應義塾大学 環境情報学部

梶原 浩紀

マイクロブログを用いた キーワードと地理的位置の対応付けシステム

近年、携帯端末で位置情報を容易に利用出来る環境が整い、これらを利用したサービスが普及しつつある。特にマイクロブログは位置情報との親和性が高く、携帯端末の位置情報を利用することで自分の現在位置を簡単に他人に知らせることができるようになった。

マイクロブログの特徴は文字数制限があり、短時間で高頻度に投稿できることである。この特徴から、本研究ではイベントに参加した人が位置情報付きでマイクロブログに投稿するという場面が増加していると考えた。

また、イベントは同じキーワードで表されるものであっても、夏の花火大会のように毎週開催場所が変わるようなものもある。例えば、人が使用する「花火」というキーワードが毎週別の場所を指し示す場合があるのではないかと考えた。

本研究の目的はマイクロブログの位置情報付きテキストを分析することで、地理的位置とキーワードを対応付けることである。本論文ではマイクロブログのテキストを収集し、データベースに登録するシステムと、これを検索して地図上にマッピングする Web アプリケーションを提案・実装・評価した。

その結果、提案する手法を用いて現実世界の地理的場所とキーワードを対応付けることができ、キーワードが指し示す地理的位置の時間による移り変わりも確認することができた。本研究の成果は携帯端末と位置情報の組み合わせが現実世界の状況をユーザが伝える有効な手段であることを示し、マイクロブログを分析することで現実世界の地理的位置とキーワードを対応付けることができた。

キーワード

1. マイクロブログ
2. Twitter
3. 検索

Binding Keywords and Real-World Locations Using Microblog

Recently, location-based services have become used widely because of the improvement of the infrastructure that can use location information from portable computers. Location information is frequently used with microblogging services; by posting to a microblogging service along with location information of portable computers, people can express their current locations to others easily.

The representative characteristics of microblogs are limits in lengths and high frequency of post events. According to these characteristics of microblogs, it is conceivable that posts to microblogs with location information, by users who have participated to a particular event, is increasing.

Upon detecting a particular event from microblog posts, there are keywords that change their actual meanings every week by the locations of where the events take place. For example, a word "fireworks" could be used for indicating different fireworks events every week during a summer season in different locations.

The purpose of this research is to bind changes in geographic locations and keywords, which were collected and analysed from microblog posts with location information. This thesis proposes, implements and evaluates a database system that collects text data from microblogs, and a web application that points the collected data on a map.

As the result, the system has detected real-world events and their changes over time by using the proposed method. This work proved that the use of location information in microblog posts is an effective mean for binding geographic locations and keywords.

Keywords :

1 . Microblog, 2. Twitter, 3. Search

Keio University , Faculty of Environment and Information Study

Hiroki Kajihara

目次

第1章	序論	1
1.1	はじめに	1
1.2	本論文の構成	1
第2章	位置情報とソーシャルメディア	2
2.1	場所に関する表現	2
2.1.1	位置表現とキーワード	5
2.2	位置情報	5
2.2.1	インターネットにおける位置情報の概要	6
2.2.2	位置情報と携帯端末	6
2.3	ウェブログとマイクロブログ	6
2.3.1	Twitter	6
2.3.2	Twitter と位置情報	7
2.4	本章のまとめ	8
第3章	関連研究	9
3.1	Twitter を利用した関連研究・サービス	9
3.1.1	Earthquake shakes Twitter users: real-time event detection by social sensors	9
3.1.2	TPS	10
3.2	Location-Based Service	11
3.2.1	位置確認・トラッキング	11
3.2.2	位置ゲーム	12
3.2.3	クーポンサービス	14
3.3	関連研究のまとめと提案システム	15
第4章	設計	16
4.1	システム図	16
4.2	Twitter Streaming API	17
4.3	データベースと形態素解析	18
4.3.1	データベース	18
4.3.2	形態素解析	18
4.4	Web アプリケーション:最大瞬間なう速システム	19
4.4.1	KML	20
4.5	本章のまとめ	21
第5章	実装	22
5.1	実装環境	22
5.2	データベース	22
5.2.1	重複の回避	22
5.2.2	時刻の修正	23

5.3	Tweet の取得	23
5.4	データベースへの格納	25
5.5	検索	26
5.5.1	PHP スクリプト	26
5.6	マッピング	28
5.7	本章のまとめ	28
第 6 章	評価	29
6.1	評価方針	29
6.2	評価環境	29
6.3	評価結果	29
6.3.1	センサとしての Twitter	30
6.3.2	時系列性	35
6.4	考察とまとめ	56
第 7 章	結論	57
7.1	本研究のまとめ	57
7.2	今後の課題と展望	57
	参考文献	60
	謝辞	61

目次

2.1	Google カレンダーの予定登録画面	2
2.2	Google カレンダーでの地図リンク	3
2.3	Google カレンダーが正しく場所を理解できた例	4
2.4	Google カレンダーが誤って場所を理解した例	4
2.5	2010 年関東花火マップ [1]	5
2.6	Twitter ホーム画面の一部	7
3.1	地震に関する Tweet をプロットし、震源地を予測した図 [2]	9
3.2	2011 年の日本の TPS を視覚化した動画の一部 [3]	10
3.3	iPhone3GS の地図アプリケーション	11
3.4	NTTdocomo 社が販売するキッズケータイ [4]	12
3.5	foursquare の web ページ	13
3.6	iPhone 版イマナラ!の画面	14
4.1	全体のシステム図	16
4.2	MeCab での形態素解析の例：日本語	18
4.3	MeCab での形態素解析の例：英語	19
4.4	最大瞬間なう速システムトップページ	19
4.5	kml のサンプル	20
5.1	MySQL のデータベース作成コマンド	22
5.2	MySQL のフィールド	23
5.3	MySQL の重複フィールドの削除	23
5.4	MySQL で GMT から JST に変換する SQL	23
5.5	Ruby での Streaming API の取得	24
5.6	取得した Tweet を Ruby で MySQL へ格納	25
5.7	検索システム図	26
5.8	PHP による MySQL への問い合わせ	27
5.9	PHP による KML の生成	27
5.10	アプリケーションのインターフェース	28
6.1	「新幹線」というキーワードで検索した結果	30
6.2	新幹線路線図 JR グループ Web サイトより	30
6.3	「山手」というキーワードで検索した結果	30
6.4	山手線路線図	30
6.5	「AKB」というキーワードで検索した結果	31
6.6	箱根駅伝コース 読売新聞社 Web サイトより	31
6.7	往路のスタートからゴールまでの時間、「箱根」で検索した結果	32
6.8	復路のスタートからゴールまでの時間、「箱根」で検索した結果	33
6.9	2 日間往路スタートから復路ゴールまでの時間、「箱根」で検索した結果	33
6.10	箱根駅伝前日の、「箱根」で検索した結果	34

6.11	1 区の時間帯に「箱根」で検索した結果	36
6.12	1 区コース	36
6.13	2 区の時間帯に「箱根」で検索した結果	37
6.14	2 区コース	37
6.15	3 区の時間帯に「箱根」で検索した結果	38
6.16	3 区コース	38
6.17	4 区の時間帯に「箱根」で検索した結果	39
6.18	4 区コース	39
6.19	5 区の時間帯に「箱根」で検索した結果	40
6.20	5 区コース	40
6.21	6 区の時間帯に「箱根」で検索した結果	41
6.22	6 区コース	41
6.23	7 区の時間帯に「箱根」で検索した結果	42
6.24	7 区コース	42
6.25	8 区の時間帯に「箱根」で検索した結果	43
6.26	8 区コース	43
6.27	9 区の時間帯に「箱根」で検索した結果	44
6.28	9 区コース	44
6.29	10 区の時間帯に「箱根」で検索した結果	45
6.30	10 区コース	45
6.31	1 区の時間帯に「駅伝」で検索した結果	46
6.32	1 区コース	46
6.33	2 区の時間帯に「駅伝」で検索した結果	47
6.34	2 区コース	47
6.35	3 区の時間帯に「駅伝」で検索した結果	48
6.36	3 区コース	48
6.37	2 区の時間帯に「駅伝」で検索した結果	49
6.38	4 区コース	49
6.39	5 区の時間帯に「駅伝」で検索した結果	50
6.40	5 区コース	50
6.41	6 区の時間帯に「駅伝」で検索した結果	51
6.42	6 区コース	51
6.43	7 区の時間帯に「駅伝」で検索した結果	52
6.44	7 区コース	52
6.45	8 区の時間帯に「駅伝」で検索した結果	53
6.46	8 区コース	53
6.47	9 区の時間帯に「駅伝」で検索した結果	54
6.48	9 区コース	54
6.49	10 区の時間帯に「駅伝」で検索した結果	55
6.50	10 区コース	55

表 目 次

4.1	Twitter API の比較	17
4.2	データベースフィールド	18
5.1	実装環境	22
6.1	早稲田大学のタスキ中継時間	32

第1章 序論

1.1 はじめに

近年、ソーシャルメディアが注目を浴びている。国内外の様々なサービスがあり、利用者は増加している。ソーシャルメディアは人と人を現実世界以外でも繋げるサービスとして、人がメッセージをやり取りするツールとなっている。

これに加え、スマートフォンの普及も目立ってきた。スマートフォンの特徴はいつでもどこでも手軽にインターネットを利用できることである。多種多様なアプリケーションが用意されていて、スマートフォンが1台あれば電話・電子メール・インターネットブラウズ等、様々なサービスを利用することができる。

スマートフォンの普及に伴って、位置情報を利用したサービスへの関心が高まってきた。スマートフォンには全地球測位システム (Global Positioning System:GPS) によって現在位置を取得できる機能がある。取得した位置情報を利用することで現在位置を地図アプリケーションで確認したり、現在地から目的地までの経路をナビゲートするアプリケーションを利用したりできる。

ソーシャルメディアと携帯端末を組み合わせることによって、今まで机に着いて時間をかけてウェブログを書いて表現をしていたものを外出先で思いつくままに手軽に投稿できるようになった。文字数が短く制限されたマイクロブログの登場とスマートフォンの普及により、外出先でソーシャルメディアを利用することはますます一般的になりつつある。スマートフォンによって取得した位置情報をソーシャルメディアの投稿に付加することで自分の現在位置を簡単に他人に知らせることができるという新しい利用方法も登場した。

本研究はこのような、位置情報付きのマイクロブログへの投稿を収集・分析することで、キーワードと地理的位置を対応付けることが目的である。

1.2 本論文の構成

本論文は全7章で構成される。第1章で本研究の背景と目的について述べ、この背景をさらに説明するために第2章でインターネットと位置情報の関係、及びマイクロブログを含むソーシャルネットワーキングについて説明する。第3章では関連する研究や技術について論じ、既存研究の手法や問題点を述べ、第4では提案手法を実装するにあたっての設計について述べる。第5章では提案手法の実装についてコードを記述しながら詳細に述べる。第6章では実装したシステムの評価方針、評価方法、評価結果、考察を述べ、第6章で本研究で提案したシステムを評価した結果から得られる結論を述べる。

第2章 位置情報とソーシャルメディア

この章では本研究の前提となる技術・概念である位置情報とマイクロブログ等のソーシャルメディアについて論じ、これらの技術とサービスの活用例について示す。

2.1 場所に関する表現

本節ではスケジュールを例として人が位置情報を扱う場合、どのような表現を用いているかを論じる。

人がスケジュールに予定を登録する際には、日付、件名、時間、場所等を記入する。その際、場所には図 2.1 のように「大学」や「職場」といった表現が使われることが多い。なぜなら、いつも通学している大学についてわざわざ「慶應義塾大学湘南藤沢キャンパス」などと入力する必要はないからである。

このスクリーンショットは、Google カレンダーの予定登録画面を示しています。画面の上部には「◀ カレンダーに戻る」ボタンと「保存」「破棄」ボタンがあります。イベントのタイトルは「授業」です。日付は「2011/01/01」から「2011/01/01」までで、「タイムゾーン」を選択しています。繰り返し設定は「終日」が選択されています。下部には「場所」のテキストボックスがあり、「大学」と入力されています。このテキストボックスは赤い枠で囲われています。また、「外部向け表示」や「プライバシー」などの設定項目も確認できます。

図 2.1: Google カレンダーの予定登録画面

2.1. 場所に関する表現

しかし、「大学」や「職場」といった表現が実際にどこの場所を示しているのかは予定を登録したユーザにしかわからず、コンピュータがこれを正確に判断することは難しい。例えば、Google カレンダー [5] の「場所」項目に場所に関する言葉を入れると、図 2.2 のように地図を表示させるリンクが出現する。このリンクをクリックすることで Google マップで場所を確認することができる。図 2.3 は「場所」に「慶應義塾大学湘南藤沢キャンパス」を指定した時の Google マップ [6] の表示例である。この場合、詳細に場所を指定しているので Google マップは正常に慶應義塾大学湘南藤沢キャンパスにピンを打つことができる。

また、図 2.4 の例な事態も発生し得る。この例はスケジュール登録時に場所に「大学」と入力したものであるが、Google カレンダーはユーザの入力した「大学」がどこか特定することができず、日本全国の「大学」と付く場所の検索結果を表示してしまっている。

この原因はユーザの使う曖昧な場所表現がコンピュータに理解できないことである。この例によって示されるように、人が使う場所に関する表現はしばしば曖昧なものになる。このような曖昧な表現は本人や共通の認識を持った人同士であれば簡潔で便利な表現であるが、コンピュータにとっては曖昧すぎて正確な場所がどこかわからないという問題点がある。



The image shows a screenshot of the Google Calendar event creation interface. At the top, there is a title field containing '卒論本提出'. Below the title, there are date and time selection fields: '2011/01/19 ~ 2011/01/19' and a 'タイムゾーン' (Time Zone) link. There are checkboxes for '終日' (All day) and '繰り返し...' (Repeat...). Below these are buttons for '予定の詳細' (Event details) and '時間を探す' (Find time). The '場所' (Location) field contains '慶應義塾大学湘南藤沢キャンパス' and has a red box around the '地図' (Map) link. The 'カレンダー' (Calendar) dropdown is set to 'かじの予定' (Kaji's calendar). There is an empty '説明' (Description) field. Below the location field, there are notification settings: '通知' (Notification) set to 'ポップアップ' (Pop-up), '10 分' (10 minutes), and a '通知を追加する' (Add notification) link. At the bottom, there are options for '外部向け表示' (External display) with radio buttons for '予定なし' (No event), '予定あり' (Event), and 'プライバシー' (Privacy) with radio buttons for 'デフォルト' (Default), '一般公開' (Public), and '限定公開' (Private). There are also links for '予定の限定公開/一般公開の詳細' (Details of limited/public events) and '予定を公開' (Share event).

図 2.2: Google カレンダーでの地図リンク



図 2.3: Google カレンダーが正しく場所を理解できた例



図 2.4: Google カレンダーが誤って場所を理解した例

2.2. 位置情報

2.1.1 位置表現とキーワード

本節では花火大会を例に位置情報とキーワードについて述べる。花火大会は夏によく行われるイベントであり日本全国様々な場所で開催される。例えば2010年の7月、8月は毎日のようにどこかで花火大会が行われていた [7]。図 2.5 は2010年の夏期主要花火大会の地図である。2010年7月27日に第44回葛飾納涼花火大会 [8] が、2010年7月31日に第33回隅田川花火大会 [9] が行われた。

それぞれ主な会場は葛飾区柴又野球場と隅田川流域であった。この時、人が使う「花火」というキーワードが指し示す場所は7月27日の場合は柴又区野球場、7月31日の場合は隅田川を示している可能性が高い。「花火」という言葉は本来、場所を示す言葉ではないが、場所を表す言葉として人に使われることもあり得る。

このように、何らかのイベントが行われている時、人はこのイベントを表すキーワードを使い、そのキーワードは同一であっても用いられる文脈と時刻によって指し示す場所が違ふことがあり得る。

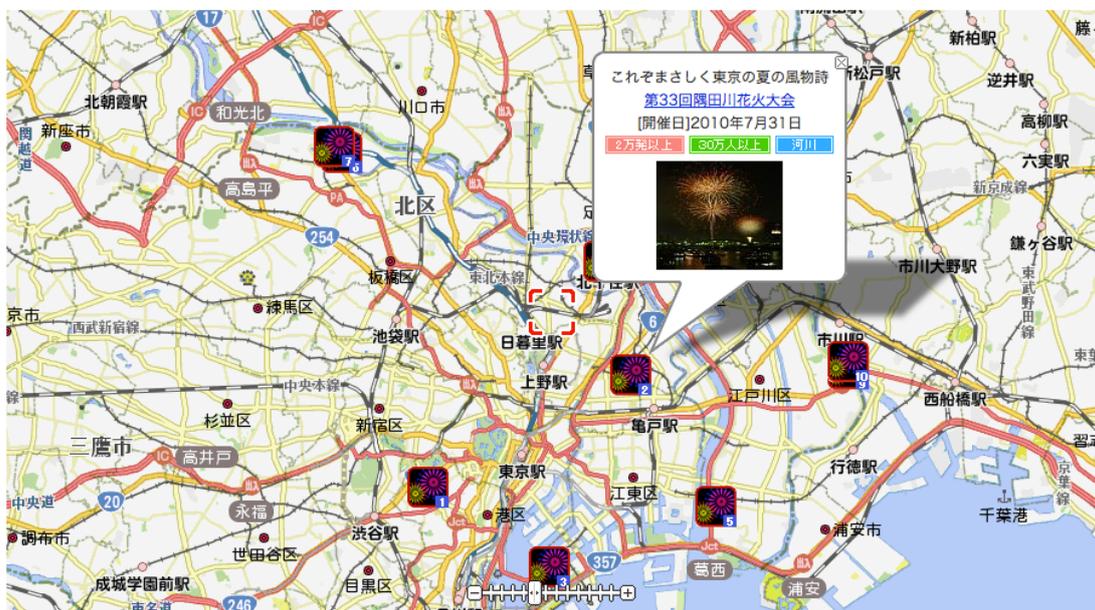


図 2.5: 2010 年関東花火マップ [1]

2.2 位置情報

本研究における位置情報とは、「緯度経度によって表すことのできるヒトやモノの場所」と定義する。位置情報を取得するにはGPSを利用するのが一般的である。自分の緯度経度が判れば地球上のどこにいても自分がの現在位置を知ることができる。

2.2.1 インターネットにおける位置情報の概要

インターネットと位置情報を組み合わせることによって、人は簡単に自分の現在位置を把握できるようになった。スマートフォンの GPS を使用することで緯度経度情報を取得し、インターネットから地図をダウンロードして自分の位置を知ることができるようになった。また、ナビゲーションアプリケーションを利用すれば GPS で取得した現在地を元に自分の周りの店舗等を検索し、経路案内サービスを受けることができる。携帯端末、とりわけスマートフォンと GPS を利用すれば場所に関する様々なサービスとそれを提供するアプリケーションを開発することができる。以下にその概要とサービスの例を挙げていく。

2.2.2 位置情報と携帯端末

総務省が 2006 年 1 月に改正、2007 年 4 月に施行した事業用電気通信設備規則 [10] により全ての 3G 携帯電話に GPS モジュールの内蔵が義務付けられた。この規則改正の目的は対応端末から 110 番や 119 番通報をしたときに通報者の位置情報を GPS によって測位し警察や消防に自動通知させることである。この改正により携帯端末には GPS モジュールが付加され、緊急通報意外にも様々な用途に活用されている。

2.3 ウェブログとマイクロブログ

ウェブログはウェブ上に記事を投稿できるサービスである。日々あったことを日記のように投稿したり、ニュース解説や政治的主張をする記事を投稿したりして自分の意見を世界に発信することができる。ウェブログには記事を読んだ人がコメントを書き込める機能があり、読者とコミュニケーションが取れる。ウェブログの利用者は 2009 年の総務省調べによると 2009 年 1 月末で約 2,695 万人で、同月の月間閲覧数は約 205 億である [11]。今では人が自分を考えを文章でインターネットで表現する時に最も人気のあるサービスであると言える。

近年、ウェブログの投稿文字数を制限したマイクロブログの利用者が増え始めている。マイクロブログは自分の現在の状況やふと考えたことを短い文章で投稿するスタイルのブログである。ある人の投稿へ返信するという機能もあり、短いテキストであるためにチャットのような使われ方がされている場面もある。

ウェブログとマイクロブログの共通点は自分の考えを表現でき、これに対するメッセージのやりとりができるという点である。これに対し両者の異なる点は、投稿文字数と一日の投稿頻度である。各ウェブログサービスの文字制限は最大全角 2 万文字、少なくとも 5000 文字である [12] のに対して、ミニブログは 140 文字から 400 文字程度の文字制限が掛けられている。マイクロブログは文字制限があり投稿文字数が少ないため手軽に思ったことを投稿することができる。また、投稿に対しても短い文字数で返信をするためチャットのような状態になることもあり、高頻度にメッセージのやりとりが行われている。本研究ではマイクロブログでも特に登録者数が多い Twitter [13] を取り上げる。

2.3.1 Twitter

Twitter [13] は 140 文字の文章 (Tweet) を投稿するマイクロブログサービスである。Twitter は 2010 年 4 月時点で 1 億 600 万以上の登録アカウント数 [14] があり、さらに毎日 30 万人ものユーザが増加している。

ソーシャルネットワーキングサービスの性格もあり自分の家族、友人、同僚、興味のある企業等

2.3. ウェブログとマイクロブログ

と双方向のやりとりができる。また利用出来るデバイスが多岐に渡り、デスクトップから携帯端末まで広く利用できる環境が整っている。

Twitter は Web サイトが用意されてはいるが多くのユーザは携帯電話やスマートフォンの専用クライアントを利用して Tweet を投稿している [14]。携帯電話やスマートフォンには 2.2 で述べたように GPS 機能が付いているため、自分が今いる場所の緯度経度を取得して tweet と共に簡単に投稿できる。



図 2.6: Twitter ホーム画面の一部

2006 年 7 月に開始された Twitter のサービス (日本語版は 2008 年 4 月) は当初は他のユーザの発言が自分のタイムラインに表示されてはならず [15]、ただ「いまどうしてる?」という質問に応える形式の簡易ブログだけの存在であった。さらに、ある人が他人の tweet に反応したいと思った時、「@username」という形で返信をし始めた。Twitter 側もこれを公式の機能として実装した [15]。この機能により、Twitter はソーシャルネットワーキングとしての性格が出始めた。

アメリカ大統領選挙では現大統領のオバマ氏が Twitter と YouTube を使用して選挙戦を行ったり、[16] イランの大統領選挙をめぐる市民の抗議活動の声は Twitter を通して世界に発信された。 [17]

2.3.1 章で述べたように、Twitter のユーザ数は増え続けており、これからますます Twitter の影響力は増していくと考えられる。

2.3.2 Twitter と位置情報

Twitter 社は 2010 年 3 月、ジオタグによって自分の詳しい位置を表示させる機能を実装した [18]。携帯デバイスの Twitter クライアントアプリケーションは tweet 投稿時に緯度経度を付加できる機能を実装し、手軽にジオタグを付けることができるようになった。Twitter ユーザにとってはジオタグを付けることで自分が実際にその場所にいた事の証明にもなる上、後々地図で確認すれば自分の行動履歴を知ることができる。

クライアントアプリケーションによってはジオタグを利用して、自分の近くにいる Twitter ユーザを検索できるものがあり、ユーザ同士のコミュニケーションを支援するひとつの機能となっている。

2.4 本章のまとめ

人が場所を言い表す時、正確な住所や緯度経度ではなく極めて個人的で曖昧な表現を用いることが多い。この表現は人間同士のコミュニケーションならば簡潔で便利であるが、コンピュータには曖昧すぎて正確な場所を特定することが難しい。また、場所表現は曖昧であると同時に、用いられる状況と時刻によって同じ人物でも違う場所を指し示すことがあるという側面もある。

また、人が緯度経度を扱う場面が増えつつある。携帯端末に GPS が搭載され、自分の現在位置を取得し様々なサービスを受けられるようになった。

携帯端末はインターネットへのアクセスを場所に関係なく可能にし、これによって近年普及したソーシャルメディアの利用が加速した。中でも、マイクロブログはその手軽さから高頻度に投稿され、携帯端末によって取得した緯度経度を付加して他人に自分の場所を知らせるような利用方法も登場した。

位置情報・携帯端末・ソーシャルメディアの親和性は非常に高いといえ、様々なサービスや利用方法が登場し、これからも増え続けると思われる。

第3章 関連研究

本研究では Twitter と位置情報を組み合わせた既存の研究について述べる。第2章で述べたような Twitter の多数のユーザ数と社会性を利用した様々な研究が行われている。

3.1 Twitter を利用した関連研究・サービス

Twitter は2006年に誕生した新しいサービスであるが、そのユーザ数の多さ、世界的な利用によって研究対象としての価値を十分に持っている。本節では Twitter に関連する関連研究を紹介した後、本研究の立ち位置を述べる。

3.1.1 Earthquake shakes Twitter users: real-time event detection by social sensors

榊らは Twitter を監視することで地震や台風などの事象が起こったことを検知する手法を研究した [2]。

Twitter を初めとするマイクロブログの特徴は即時性である。一般のブログはひとつの記事を投稿するのに一日から数日かかるが、マイクロブログは1日に何度も投稿されるのが特徴である。榊らはこの特徴に着目し、Twitter ユーザをイベントのセンサだとみなして地震や台風等のイベントが起こったことを検知することに利用した。

図3.1は「地震」、「揺れた」を含む Tweet から地震直後のもののみを抽出し、地震検出・震源地推定をしたものである。

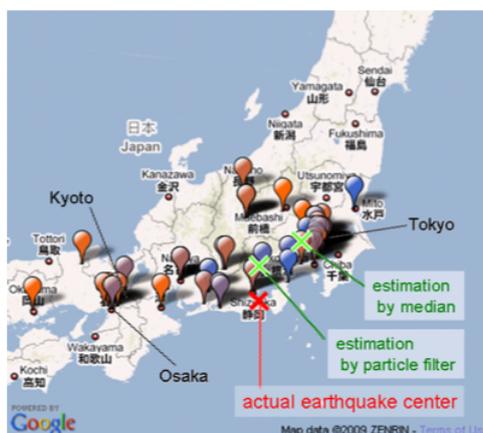


図 3.1: 地震に関する Tweet をプロットし、震源地を予測した図 [2]

3.1.2 TPS

秒間の Tweet 数を Twitter-Per-Second と呼ぶ。TPS が高いと、その瞬間に多くの人が同時にひとつのイベントについて Tweet をしたということを示す。2011 年 1 月 7 日時点での TPS 世界記録は日本が 2011 年を迎える年越しの瞬間で、新年を祝う Tweet が 6,939TPS に達した [3]。図 3.2 は Twitter 社が世界の新年を祝う Tweet の TPS を視覚化した動画の一部である。その他にも、2010 年サッカーワールドカップ南アフリカ大会 [19] の際に記録した 3,283TPS がある。これは日本が決勝トーナメント進出を決めた瞬間に記録されたものである。日本時間午前 3 時過ぎの試合開始であるにも拘らずテレビ中継の視聴率は 40%を超えた試合 [20] であり、注目度が高いイベントは TPS に大きく影響するということを示している。

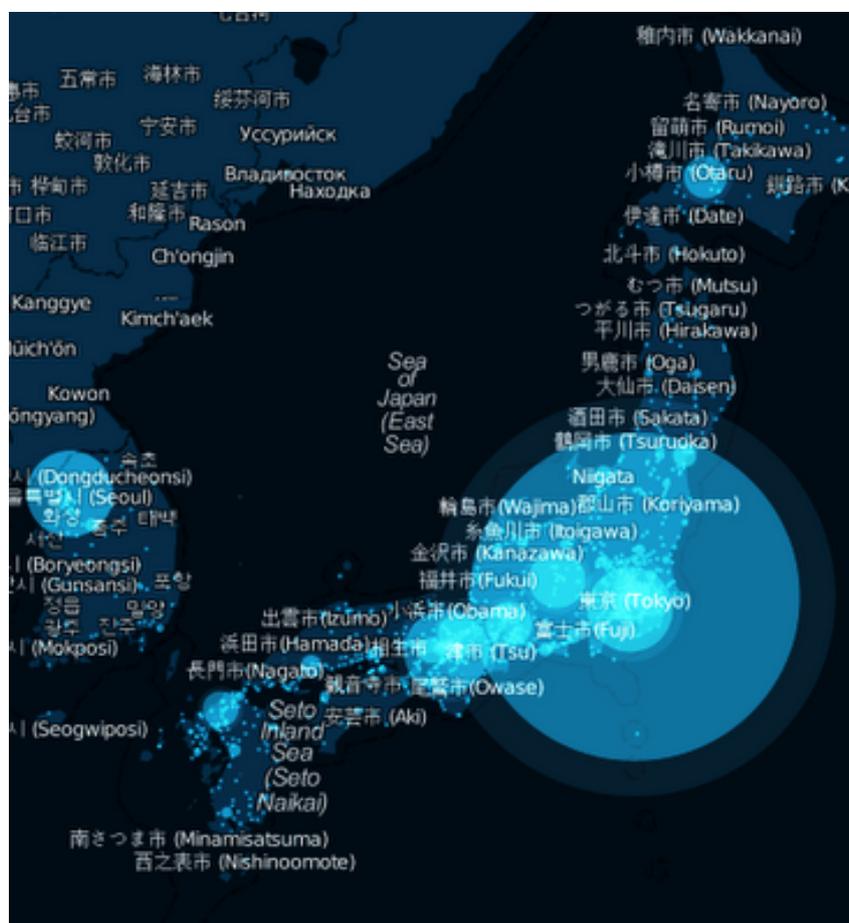


図 3.2: 2011 年の日本の TPS を視覚化した動画の一部 [3]

3.2 Location-Based Service

位置情報を利用したサービスを Location-Based Service と呼び、自分の位置を他人に知らせるものから、ゲームまで様々なものがある。位置情報をスマートフォン等に搭載された GPS を利用して取得し、サービスに利用している。人とのコミュニケーションを支援するものから商業的な販売促進に繋がるサービスまで様々なものが登場し、これからも成長が予想されている分野である。本節では Location-Based Service について分野別に例を挙げ、位置情報を扱うことがどれだけ一般の携帯端末ユーザに浸透しているかを示す。

3.2.1 位置確認・トラッキング

iOS4 の地図アプリケーション

Apple 社 [21] が開発した iPhone [22] に搭載されている iOS4 [23] の地図アプリケーションは地図をインターネットを通じてダウンロードして表示する。図 3.3 は地図アプリケーションを利用して経路を検索した例である。



図 3.3: iPhone3GS の地図アプリケーション

キッズケータイ

図 3.4 に示すキッズケータイ [4] は NTT ドコモが販売する子ども用携帯電話である。保護者は子どもに持たせたキッズケータイの位置情報をメールで受け取ったり、自分の携帯電話の Web ブラウザで確認したりすることができる。



図 3.4: NTTdocomo 社が販売するキッズケータイ [4]

ケータイ版ナビタイム

ナビタイム [24] は電車の乗り換え案内や自動車のルート検索サービスを提供するサービスである。携帯電話で取得した位置情報を利用して利用するのがケータイ版ナビタイムで、このサービスを利用することで利用者はカーナビのような音声案内サービスを受けることができる。これまでカーナビでしか提供されてこなかった経路案内サービスが、GPS の一般への普及によってスマートフォンや携帯電話で利用出来るようになった。

3.2.2 位置ゲーム

携帯端末の GPS を利用したゲームも人気があり、様々なものが登場している。典型的な遊び方としては、自分が今いる緯度経度を登録することでその場所を登録した他のユーザと場所を取り合ったり、登録した場所や数に応じたアイテムを得ることができたりするものがある。日本ではケータイ国盗り合戦 [25] や foursquare [26] が有名である。位置ゲームに関しては第 3.2 章にて詳しく述べる。

foursquare

foursquare [26] は位置情報を利用した SNS・位置ゲームである。店舗や駅など、自分が現在いる場所 (check-in) を登録していくことでポイントや様々な種類のバッジが取得できる。特定の場所

3.2. LOCATION-BASED SERVICE

に頻繁に通って登録することでその場所の「メイヤー」の称号を得ることができるなど、ゲーム性もある。check-inする度にポイントが付く仕組みとなっていて、このポイントを集めることで現実の店舗の割引クーポンとなる [27] 等、現実世界との連携もなされている。実際に、海外ではスターバックスやマクドナルドなどの店舗が foursquare の会員向けに割引クーポンを発行している [28]。



図 3.5: foursquare の web ページ

3.2.3 クーポンサービス

イマナラ！

イマナラ!^[29] は株式会社ロケーションバリュー ^[30] が運営するクーポンサイトである。自分が今いる場所を取得して検索すると、自分が今いる場所の周りの店舗が出しているクーポンを閲覧できる。クーポンには制限時間が付いていて、クーポンをダウンロードして制限時間内に店舗に到着すれば割引を受けられる。店側は席に空きがあったり、閉店まで間もない場合であったりする等、閑散期にだけクーポンが出せるといったサービス展開ができる。ユーザは自分が今いる場所の近くの店舗でクーポンが出ている店を探せるという利点がある。

このサービスはスマートフォンとの連携が欠かせないものである。スマートフォンの GPS で自分の現在位置を取得し、図 3.6 のように自分から近い位置の店舗からの割引情報を受け取ることができる。アプリケーションをバックグラウンドで起動しておくことで自分の現在位置を常時取得し、周辺の店舗が割引チケットを発行すると通知するなどのサービスがある。



図 3.6: iPhone 版イマナラ!の画面

3.3 関連研究のまとめと提案システム

関連研究と関連サービスによって、位置情報を利用したサービスの概要を示した。本研究で提案するシステムは Twitter への位置情報付きの投稿を収集し、キーワードを検索することで現実世界の地理的位置とキーワードの相関を視覚的に表現するものである。

位置情報を利用することで他人に自分の位置情報を伝えることができたり、様々なサービスを受けられることを本章で述べた。本研究はこれまでに述べてきた、場所に関する曖昧な表現の問題を踏まえた上で、携帯端末とソーシャルメディアの手軽さと高頻度な利用という特徴を利用した、キーワード検索システムを提案する。

第4章 設計

4.1 システム図

提案手法を図 4.1 に示す。ユーザが投稿したメッセージは Twitter Streaming サーバに流される。Twitter クローラシステムがこの Streaming サーバに接続し、位置情報付きの Tweet を収集する。収集したデータは DB 登録モジュールがデータベースに登録する。このとき、DB 登録モジュールは検索インデックスとなるキーワードを形態素解析によって生成しデータベースに登録する。

イベントを検索するための Web アプリケーションで検索キーワードを入力すると、データベース検索モジュールがデータベースにクエリを送信し、データベースからの応答を元に地図にマッピングし、検索結果を表示する。

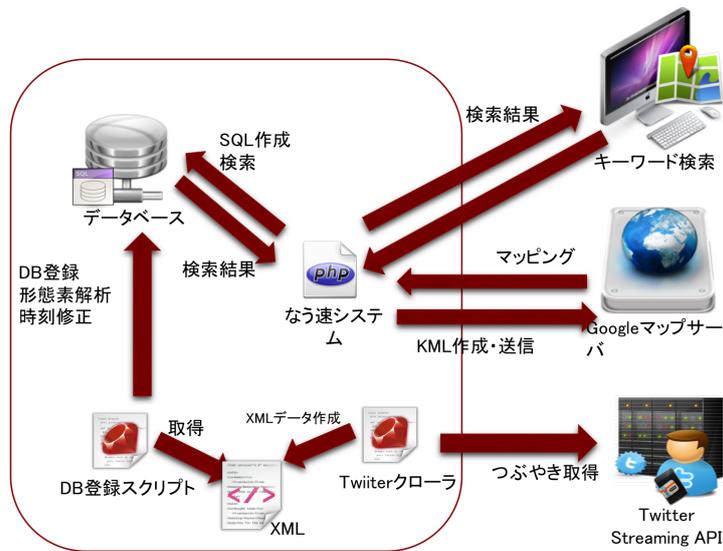


図 4.1: 全体のシステム図

4.2 Twitter Streaming API

Twitter には表 4.1 のように 4 つの Tweet 取得のための API が用意されている。Streaming API の 3 つは Tweet が常に流れてくる Timeline(TL) というサーバに接続して継続して Tweet を取得するための API である。これに対して http リクエストを送る API もある。これは主に Twitter をブラウザ以外のデスクトップクライアントやスマートフォンクライアントで利用する際に使用する API で、60 分間に 150 回までの取得という制限がある。本システムでは継続して Tweet を取得することが必要であるため、Streamin API を選択した。

Streaming API は 3 つの Tweet 取得レベルに分けられている。図 4.1 にある Sprizer は全 tweet の 1%、Gardenhose は 10% を取得できる。Firehose は全 Tweet を取得できる API であるが、一般のデベロッパには解放されておらず、Twitter 社と特別な契約をすることが必要 [31] である。

本システムでは利用可能な Streaming API の中で取得精度の最も高い Gardenhose を使用する。

表 4.1: Twitter API の比較

種類	取得精度	利用制限
StreamingAPI Spritzer	全 tweet の 1%	なし
StreamingAPI Gardenhose	全 tweet の 10%	なし
StreamingAPI Firehose	全 tweet 取得可	利用不可
http GET リクエスト	public_timeline の取得	60 分間に 350 回

4.3 データベースと形態素解析

Twitter API で取得した Tweet は xml 形式である。このままでは扱いにくいので、Web アプリケーションで検索するためデータベースを構築する必要がある。xml から必要なパラメータを抜き出し格納する目的でデータベースを設計する。

4.3.1 データベース

提案するシステムには、Tweet 内容・投稿日時・緯度・経度と、検索インデックスが必要であるので、図 4.2 のようにテーブルを設計した。

表 4.2: データベースフィールド

word	lon	lat	date	text
インデックス	経度	緯度	投稿日時	Tweet 内容

word カラムには形態素解析後の単語を、lon カラムには経度を、lat カラムには緯度を、date カラムには tweet の投稿日時を、text カラムには取得した tweet の全文をそれぞれ格納する。

4.3.2 形態素解析

データベースを検索する際に、イベントに関連するキーワードを用いなければならない。このため、Twitter への投稿内容を形態素解析してインデックスとしてデータベースに保存する。形態素解析とは、図 4.2.4.3 のように言葉を文法や辞書を元にして最小単位に分割することである。

```
dhcp-143-232:b_thesis kaji\$ mecab
すもももももものうち
すもも 名詞,一般,*,*,*,*,すもも,スモモ,スモモ
も 助詞,係助詞,*,*,*,*,も,モ,モ
もも 名詞,一般,*,*,*,*,もも,モモ,モモ
も 助詞,係助詞,*,*,*,*,も,モ,モ
もも 名詞,一般,*,*,*,*,もも,モモ,モモ
の 助詞,連体化,*,*,*,*,の,ノ,ノ
うち 名詞,非自立,副詞可能,*,*,*,うち,ウチ,ウチ
EOS
```

図 4.2: MeCab での形態素解析の例：日本語

4.4. WEB アプリケーション:最大瞬間なう速システム

```
dhcp-143-216:~ kaji\$ mecab
The pen is mightier than the sword
The 名詞,固有名詞,組織,*,*,*,*
pen 名詞,一般,*,*,*,*,*
is 名詞,固有名詞,組織,*,*,*,*
mightier 名詞,一般,*,*,*,*,*
than 名詞,固有名詞,組織,*,*,*,*
the 名詞,一般,*,*,*,*,*
sword 名詞,固有名詞,組織,*,*,*,*
EOS
```

図 4.3: MeCab での形態素解析の例：英語

4.4 Web アプリケーション:最大瞬間なう速システム

Streaming API によって取得した緯度経度を利用して、ある単語を検索するとその単語を含む Tweet が Google マップに表示される Web アプリケーションを作成する。検索するには検索ワードと Tweet された期間を指定できるようにする。期間を指定することで移り変わっているイベントを分かりやすく検索できるようにするためである。

この Web アプリケーションを最大瞬間なう速システムと名付ける。図 4.4 は最大瞬間なう速システムのインターフェースである。



図 4.4: 最大瞬間なう速システムトップページ

4.4.1 KML

Google マップに地理データを表示させるには KML(Keyhole Markup Language) が必要である。KML は Google アース、Google マップ、モバイル Google マップで利用できる。KML はネストされた要素や属性を含むタグ構造を使用し、XML スタンドアートをベースにしている。図 4.5 に簡単な kml の例を挙げる。

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
      at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

図 4.5: kml のサンプル

図 4.5 のような KML を Google マップに読み込ませることによって地理情報を表示させることができる。最大瞬間なう速システムでは、緯度経度の他に Tweet 投稿時間、Tweet の内容を KML によって Google マップに表示させる。

4.5 本章のまとめ

提案手法の設計をまとめると、図 4.1 のようになる。Twitter ユーザが登録した Tweet は Streaming サーバへ接続することで取得できる。この役割を担うのが Twitter クローラである。Twitter クローラは Streaming サーバへ接続し、位置情報付きの投稿を収集する役割がある。

DB 登録モジュールは、収集した Tweet をデータベースに登録する役割がある。登録する際、形態素解析によって投稿内容を単語に分け、検索インデックスとして登録していく。これらのクローラ・DB 登録モジュールは別々のプログラムであり、それぞれ手動で実行する必要がある。

ユーザは Web アプリケーションを利用してデータベースにアクセスする。Web アプリケーションにはユーザが入力した検索クエリをデータベースに問い合わせ、返されたデータをマッピングする役割がある。以上の設計の実装を第 5 章で論じる。

第5章 実装

第5章では第4章で述べた設計の具体的な実装について述べる。

5.1 実装環境

システム実装にあたり、図5.1のような構成にした。

表 5.1: 実装環境

	OS	Web サーバ	データベース	Web アプリ
ソフトウェア	Ubuntu10.04	Apache2	MySQL5	Ruby1.8.7, PHP5

5.2 データベース

データベースはMySQL5を利用し、図5.1のようにして作成し、フィールドは図5.2の通りである。

```
DROP TABLE IF EXISTS twitter;  
create table twitter( word text, lon double, lat double, date  
timestamp, text text );
```

図 5.1: MySQL のデータベース作成コマンド

5.2.1 重複の回避

サンプリングには4つのアカウントを使用した為、重複したデータがデータベースに含まれている可能性がある。データベースから重複したレコードを削除する目的で図5.3のようなSQLクエリを実行した。

5.3. TWEET の取得

```
mysql> desc twitter;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default |
+-----+-----+-----+-----+
| word | text | YES | | NULL |
| lon | double | YES | | NULL |
| lat | double | YES | | NULL |
| date | timestamp | NO | | CURRENT_TIMESTAMP |
| text | text | YES | | NULL |
+-----+-----+-----+-----+
5 rows in set (0.17 sec)
```

図 5.2: MySQL のフィールド

```
#一時的にテーブル「temp_twitter」を作っておく
#フィールド「date」と「text」をグループ化する
mysql> CREATE TABLE temp_twitter as SELECT * FROM twitter GROUP BY
    date, text;
#「temp_twitter」テーブルから重複データがなくなっていることを確認した後に
mysql> ALTER TABLE temp_twitter RENAME TO twitter;
```

図 5.3: MySQL の重複フィールドの削除

5.2.2 時刻の修正

Twitter Streaming API で取得した Tweet の投稿時間はグリニッジ標準時 (GMT) である。このため日本時間から 9 時間減算された時刻となっている。これを日本標準時 (JST) に変換するために以下の MySQL クエリを実行して 9 時間加算した。

```
mysql> UPDATE kaji_hakone SET date = (select date_add(date, interval
    9 hour));
```

図 5.4: MySQL で GMT から JST に変換する SQL

5.3 Tweet の取得

第 4.2 章で述べたように、Tweet の取得には Twitter 社が提供する Streaming API を利用する。Streaming API には Ruby を利用してアクセスする。Streaming API によって次々と送信されてくる tweet をパースして xml にして保存していく。Streaming API から流れてくる tweet の中から以下の項目を保存する。Ruby のコードは図 5.5 の通りである。

```

#!/usr/bin/ruby
# coding: utf-8

require 'net/http'
require 'uri'
require 'rubygems'
require 'json'
require 'MeCab'

USERNAME = "twitter ユーザ名"
PASSWORD = "パスワード"
#ストリーミングAPI の URL
uri = URI.parse('http://stream.twitter.com/1/statuses/filter.json')

fname = "" + Time.now.strftime("%Y-%m-%d")

//追加書き込み読み書き両用モードでオープン
f = open(fname, "a+");
Signal.trap(:TERM) { f.close; exit(0) }
Signal.trap(:INT, "IGNORE")

begin
  Net::HTTP.start(uri.host, uri.port) do |http|
    request = Net::HTTP::Post.new(uri.request_uri)
    #Basic 認証で接続
    request.basic_auth(USERNAME, PASSWORD)
    #取得する緯度経度の範囲を指定 (全世界)
    request.set_form_data({'locations' => '-171,-84,179,84'}, '&')
    http.request(request) do |response|
      raise 'Response is not chunked' unless response.chunked?
      response.read_body do |chunk|
        # 空行は無視する = JSON 形式でのパースに失敗したら次へ
        status = JSON.parse(chunk) rescue next
        #Tweet をパースして変数へ
        # 削除通知など、'text' パラメータを含まないものは無視して次へ
        next unless status['text']
        #ジオタグを含まないものは無視して次へ
        next unless status['geo']
        user = status['user']
        created_at = status['created_at']
        geo = status['geo']
        coordinates=geo['coordinates']
        lat = geo['coordinates'][0]
        lon = geo['coordinates'][1]
        f.puts "#{chunk}"
      end
    end
  end
end
rescue Timeout::Error, EOFError => e
  p e
  retry
end

```

5.4 データベースへの格納

Streaming API サーバから取得したデータはテキストデータとして一度保存される。このテキストデータをデータベースに格納するためのコードが図??である。このコードは Ruby によって記述され、テキストデータから投稿時間・投稿内容・緯度・経度を抜き出し、投稿内容を MeCab によって形態素解析してデータベースに保存するプログラムである。

```
#!/usr/bin/ruby

require 'rubygems'
require 'json'
require 'mysql'
require 'parsedate'
#MeCab のライブラリを読み込む
require 'MeCab'

nlines = 0

db = Mysql::new("localhost", "ユーザ名", "パスワード", "データベース名")
mecab = MeCab::Tagger.new()

open(ARGV[0], "r") {|file|
  while l = file.gets

    #中略
    #Tweet の内容をパースして変数へ

    node = mecab.parseToNode(text)
    while node
      begin
        feature = node.feature.split( "," )
        node_word = feature[0].strip
        next if node.stat == 2 or node.stat == 3
        db.query("insert into twitter values ('#{Mysql::quote node.surface}',
          #{lon},#{lat},\#{time}\", '#{Mysql::quote text}')")
        ensure
          node = node.next
        end
      end
    end
  }
db.close
```

図 5.6: 取得した Tweet を Ruby で MySQL へ格納

5.5 検索

データベースに格納したデータを検索して Google Map 上に表示させるため、HTML, javascript, PHP を利用して Web アプリケーションを作成した。地図データの利用と tweet のマッピングのために GoogleMaps API を利用した。

図 5.7 は検索の流れを表したものである。ユーザがダイアログボックスに入力したワードは一度 PHP スクリプトに送られ、PHP は MySQL に問い合わせる。

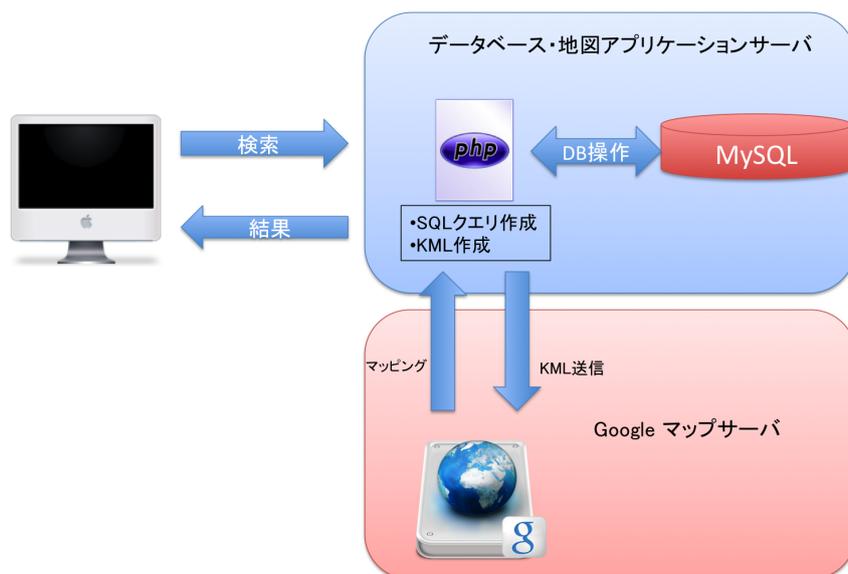


図 5.7: 検索システム図

5.5.1 PHP スクリプト

PHP スクリプトには 2 つの役割がある。1 つはユーザが入力した検索ワードを SQL にして MySQL に問い合わせることである。擬似コードを図 5.8 で示す。

2 つめの役割は、MySQL から返された結果を index.php に返信することである。返されたデータは index.php 内の javascript で kml を生成して Google マップサーバに渡される。Google マップサーバは受け取った kml を元に、ピンを地図上にドロップする。擬似コードを図 5.9 に示す。

5.5. 検索

```
//ユーザーが入力したデータを変数に格納
$word = $_GET['word'];
$date1 = $_GET['date1'];
$date2 = $_GET['date2'];

//クエリ生成
$query = "select * from twitter where date between \"\".$date1.\"\" and \"\".$date2.\"\" and word like \"\".$word.\"\" limit 0,500";

//MySQL からの返答を格納
$result = mysql_query($query);
```

図 5.8: PHP による MySQL への問い合わせ

```
downloadUrl(url, function(data) {
    var xml = parseXml(data);
    var markers = xml.documentElement.getElementsByTagName("marker");
    for (var i = 0; i < markers.length; i++) {
        var name = markers[i].getAttribute("name");
        var date = markers[i].getAttribute("date");
        var description = markers[i].getAttribute("description");
        var point = new google.maps.LatLng(
            parseFloat(markers[i].getAttribute("lat")),
            parseFloat(markers[i].getAttribute("lng")));
        var icon = customIcons["ico"] || {};
        var marker = new google.maps.Marker({
            map: map,
            position: point,
            icon: icon.icon,
            shadow: icon.shadow
        });
        bindInfoWindow(marker, map, infoWindow, "【" + date + "】" + description);
    }
});
return false;
}
```

図 5.9: PHP による KML の生成

5.6 マッピング

以上のような仮定を経て、与えられた検索ワードに対してのマッピングを行うことができる。マッピング後の様子を図 5.10 に示す。

ユーザはピンをクリックすることで吹き出しが現れ、投稿日時、投稿内容を表示させることができる。



図 5.10: アプリケーションのインターフェース

5.7 本章のまとめ

本研究の実装は Twitter Streaming API を利用して継続的に Tweet を取得できるようにした点と、取得した Tweet を視覚的にマッピングした点に注力している。

第6章 評価

第6章では、本研究の評価方針、評価環境、評価と考察を述べ、第7章の結論へと繋げる。

6.1 評価方針

本研究の評価は最大瞬間なう速システムが現実世界のイベントを検知できるか、時間と共に移り変わるイベントがあったとき、システムも時間と共に移り変わる様を検知できるかを評価し、検知できていれば本研究はイベント検知が可能であるとする。

評価の軸は以下の2点とする。これら2点の評価について、次節から詳しく述べていく。

- イベント検知の正確性
現実に行ったイベントが、正確に地図上にマッピングできているかを評価する。現実世界での場所と、地図上にマッピングされたピンとが一致しているかどうかを確認する。
- イベントが時系列に移動していくことを検知できるか
時間の遷移によって移り変わるようなイベントについて、地図上のピンも時系列に移り変わっているかどうかを見る。

6.2 評価環境

評価に際して、最大瞬間なう速システムを使用して行う。評価用に時系列に検索できるダイアログボックスを設置し、秒単位で検索ができるようにしたものを使用する。これを利用して箱根駅伝各区間の走行時間帯をそれぞれ検索し、選手の中継所通過時間と Tweet の投稿時間を比較し、両方に時間的整合性が取れ、時系列に沿って Tweet が移動しているかどうかを評価する。

6.3 評価結果

イベント検知が正確にできているかを評価するため、「新幹線」・「山手」・「AKB」という3つのキーワードを用いて検索した。また、箱根駅伝期間中に得られた Tweet を評価専用のデータベースに入れ、「箱根」、「駅伝」、総合優勝した早稲田大学を対象に「早稲田」というキーワードの3つを最大瞬間なう速システムで検索した。検索結果と解説を以下に示す。

6.3.1 センサとしての Twitter

本節では Twitter ユーザの投稿がイベント検知のセンサとして機能しているかを評価する。キーワードを検索し、マッピング結果が現実世界のイベントと整合性が取れているかを確認する。

図 6.1 は「新幹線」というキーワードで検索した結果である。この結果と図 6.2 に示した新幹線路線図を比較すると、新幹線の路線図とマッピングされたピンが一致していることが確認できる。



図 6.1: 「新幹線」というキーワードで検索した結果

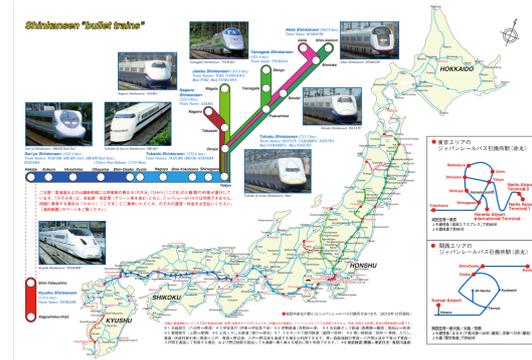


図 6.2: 新幹線路線図 JR グループ Web サイトより

図 6.3 は「山手」というキーワードで検索した結果である。山手線を利用したと思われる Twitter ユーザが投稿した Tweet が環状にマッピングされている。図 6.4 に示した山手線路線図と比較しても線路に沿ってマッピングされていることが確認できる。



図 6.3: 「山手」というキーワードで検索した結果

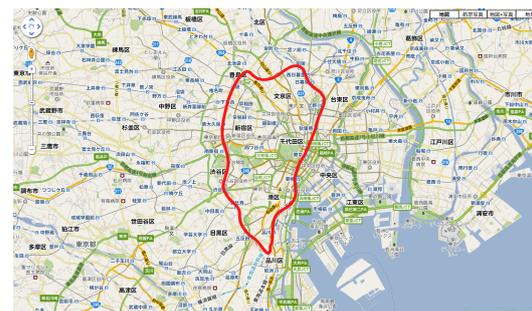


図 6.4: 山手線路線図

図 6.5 は「AKB」で検索した結果である。AKB48^[32] は秋葉原に専用ライブハウスを持つアイドルグループである。そのため、秋葉原にピンが集中していることが確認できる。また、葛西臨海公園にもピンが集中していることが確認できる。このピンを調べると全て日付が 2010 年 10 月 10 日となっている。この日は AKB48 がコンサートライブ^[33] を行った日である。ライブの観客が Twitter にライブ会場にいることを投稿した結果が検出できている。

6.3. 評価結果



図 6.5: 「AKB」というキーワードで検索した結果

図 6.7 は最大瞬間なう速システムで 2011 年 1 月 2 日に行われた往路スタートの 8 時 00 分から最下位の大学がゴールした同日 13 時 41 分までの範囲で「箱根」というキーワードで検索した結果である。図 6.6 で示したコース図に沿うように Tweet されているのが確認できる。

図 6.8 は 2 日目の 1 月 3 日に行われた復路で、スタートの 8 時 00 分から最下位の大学がゴールした 13 時 49 分までの範囲で「箱根」というキーワードで検索した結果である。この結果も往路と同様に図 6.6 で示したコース図に沿うように Tweet がマップされている。

図 6.9 は箱根駅伝 2 日間の往路スタートから復路ゴールまでの範囲で「箱根」で検索した結果である。箱根町以外でもピンが多く落ち、箱根駅伝コースに沿うように列になってマッピングされている。また、図 6.10 は 2011 年元旦の「箱根」の検索結果である。箱根町と東京都内いくつかピンが打たれているが、箱根駅伝コース上には打たれていない。



図 6.6: 箱根駅伝コース 読売新聞社 Web サイトより

表 6.1: 早稲田大学のタスキ中継時間

中継所	通過時間
大手町スタート	1月2日 8:00
鶴見中継所	1月2日 9:02
戸塚中継所	1月2日 10:10
平塚中継所	1月2日 11:13
小田原中継所	1月2日 12:09
芦ノ湖ゴール	1月2日 13:30
芦ノ湖スタート	1月3日 8:00
小田原中継所	1月3日 8:58
平塚中継所	1月3日 10:02
戸塚中継所	1月3日 11:09
鶴見中継所	1月3日 12:19
大手町ゴール	1月3日 13:29



図 6.7: 往路のスタートからゴールまでの時間、「箱根」で検索した結果

6.3. 評価結果



図 6.8: 復路のスタートからゴールまでの時間、「箱根」で検索した結果



図 6.9: 2日間往路スタートから復路ゴールまでの時間、「箱根」で検索した結果



図 6.10: 箱根駅伝前日の、「箱根」で検索した結果

6.3.2 時系列性

もう1つの評価ポイントとしてイベントが時系列に沿って移り変わることを検出することを確認しなければならない。本研究では箱根駅伝の選手の走行場所が進むに従って Tweet のピンも移り変わるかを最大瞬間なう速システムによって評価した。

図 6.11 から図 6.50 は「箱根」、「駅伝」という2つのキーワードを、図 6.1 の区間毎の時間をもとに検索した結果と、区間毎のコース図である。

両ワードとも、選手がタスキをリレーして区間が変わるのと同じように Twitter ユーザの投稿も同じく区間を移動していく様子が確認できた。2日目の往路はピンの流れが大手町から箱根芦ノ湖まで選手と同じように移り変わり、2日目の復路は箱根芦ノ湖から大手町まで選手と同じようにピンが帰っていく様子が確認でき、選手が東京箱根間を往復するのと同様にピンも往復した。

次ページより評価結果の図を載せる。



図 6.11: 1 区の時間帯に「箱根」で検索した結果



図 6.12: 1 区コース

6.3. 評価結果



図 6.13: 2 区の時間帯に「箱根」で検索した結果



図 6.14: 2 区コース



図 6.15: 3 区の時間帯に「箱根」で検索した結果



図 6.16: 3 区コース

6.3. 評価結果



図 6.17: 4 区の時間帯に「箱根」で検索した結果



図 6.18: 4 区コース

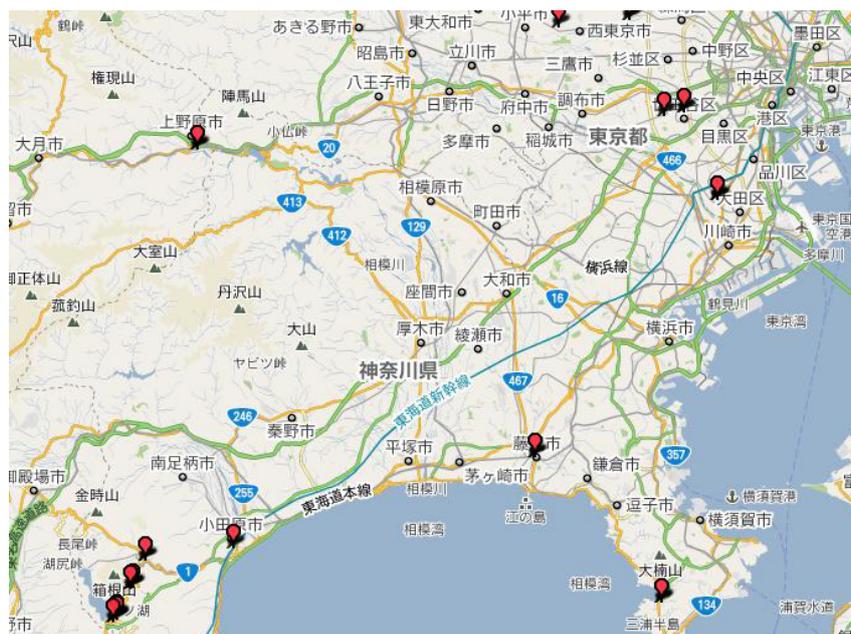


図 6.19: 5 区の時間帯に「箱根」で検索した結果



図 6.20: 5 区コース

6.3. 評価結果



図 6.21: 6 区の時間帯に「箱根」で検索した結果



図 6.22: 6 区コース



図 6.23: 7 区の時間帯に「箱根」で検索した結果



図 6.24: 7 区コース

6.3. 評価結果

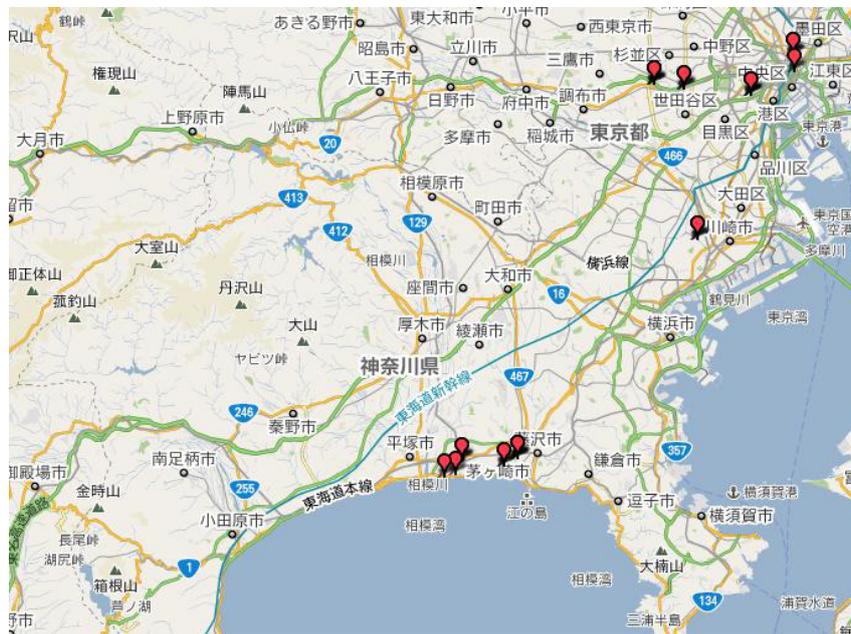


図 6.25: 8 区の時間帯に「箱根」で検索した結果



図 6.26: 8 区コース



図 6.27: 9 区の時間帯に「箱根」で検索した結果

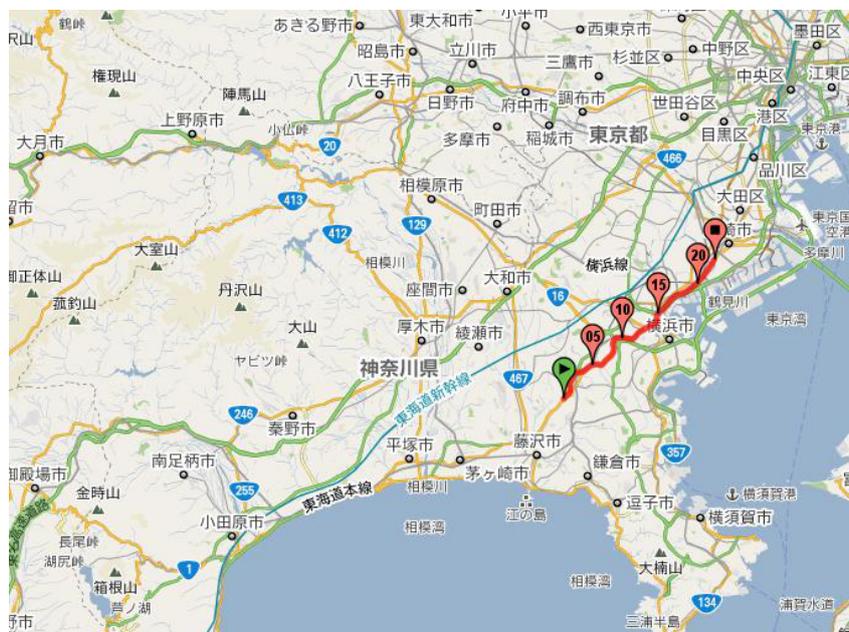


図 6.28: 9 区コース

6.3. 評価結果



図 6.29: 10 区の時間帯に「箱根」で検索した結果



図 6.30: 10 区コース

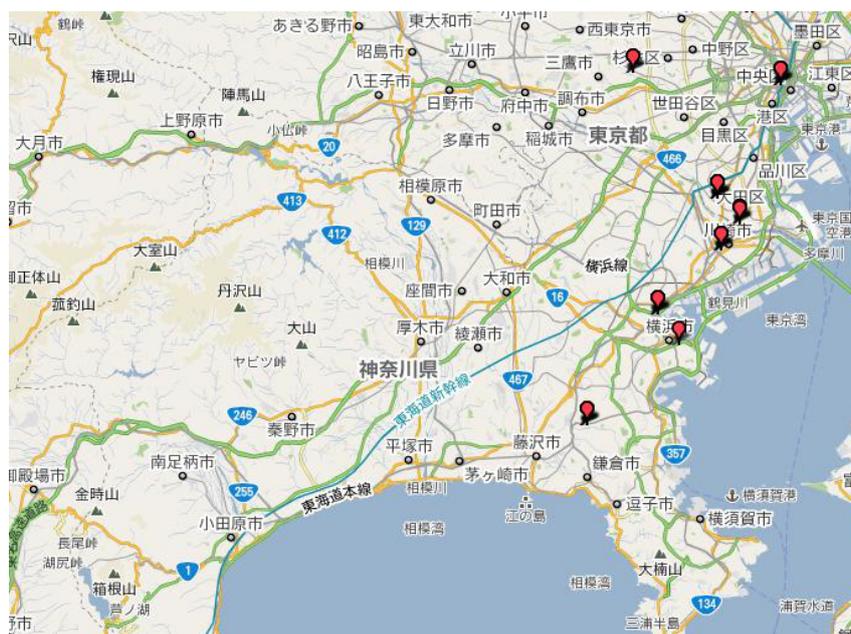


図 6.31: 1 区の時間帯に「駅伝」で検索した結果



図 6.32: 1 区コース

6.3. 評価結果



図 6.33: 2 区の時間帯に「駅伝」で検索した結果

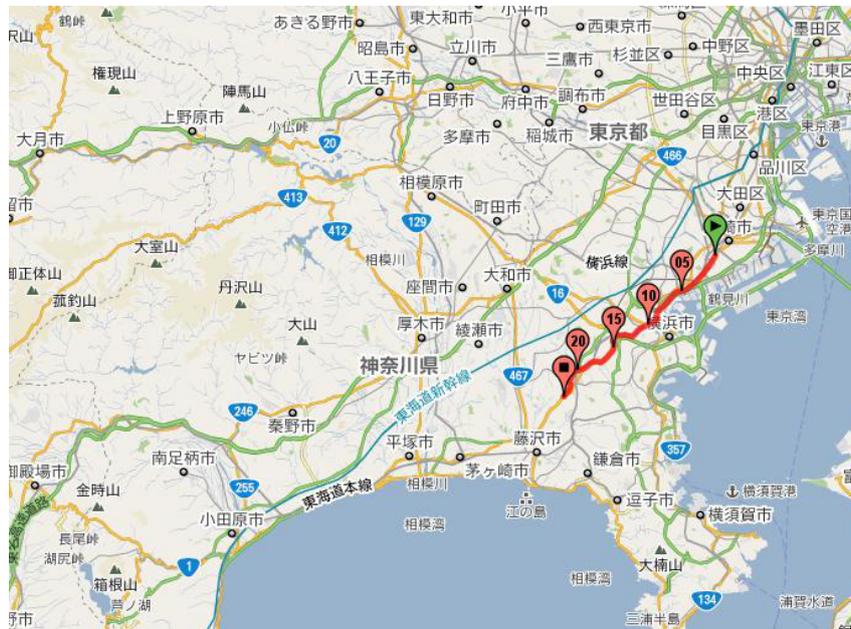


図 6.34: 2 区コース



図 6.35: 3 区の時間帯に「駅伝」で検索した結果



図 6.36: 3 区コース

6.3. 評価結果



図 6.37: 2 区の時間帯に「駅伝」で検索した結果



図 6.38: 4 区コース

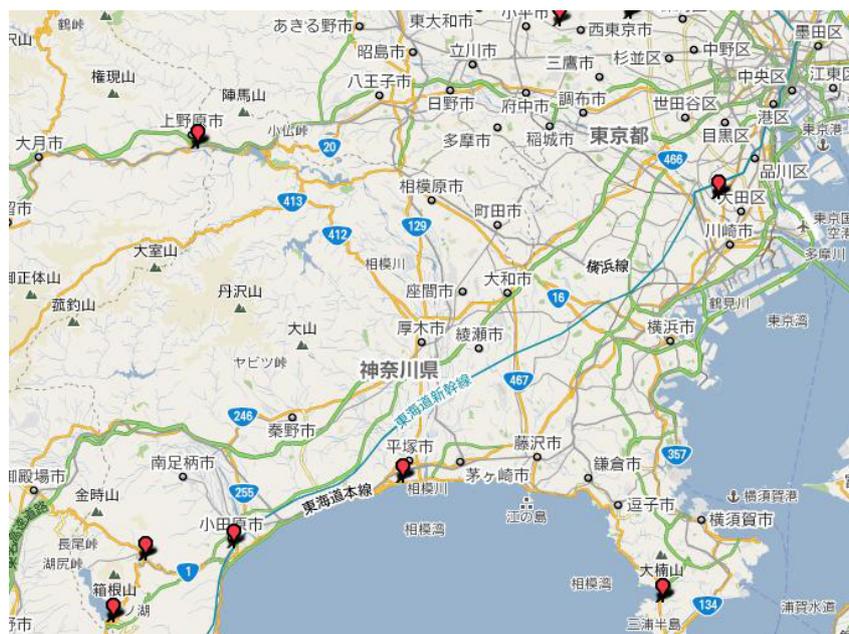


図 6.39: 5 区の時間帯に「駅伝」で検索した結果



図 6.40: 5 区コース

6.3. 評価結果



図 6.41: 6 区の時間帯に「駅伝」で検索した結果



図 6.42: 6 区コース



図 6.43: 7 区の時間帯に「駅伝」で検索した結果



図 6.44: 7 区コース

6.3. 評価結果



図 6.45: 8 区の時間帯に「駅伝」で検索した結果



図 6.46: 8 区コース



図 6.47: 9 区の時間帯に「駅伝」で検索した結果

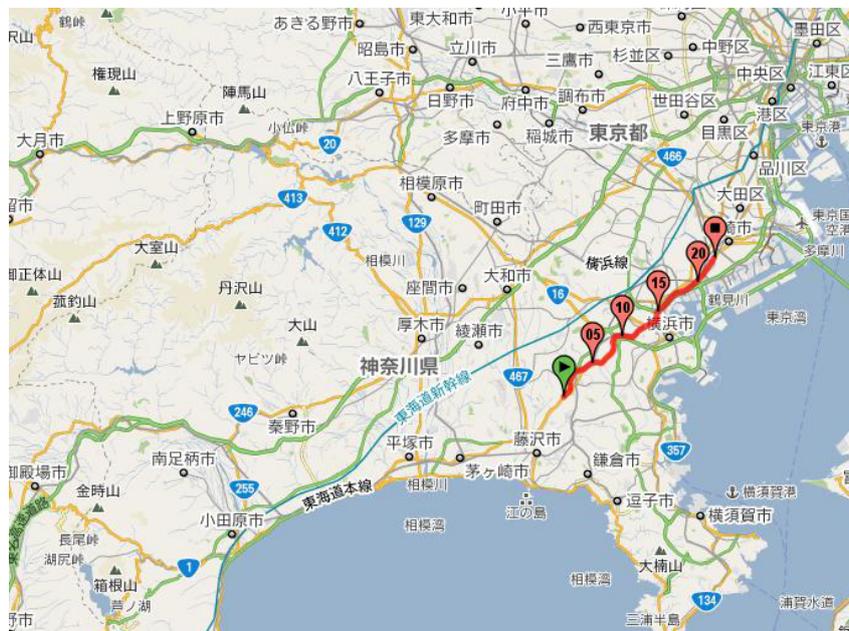


図 6.48: 9 区コース

6.3. 評価結果

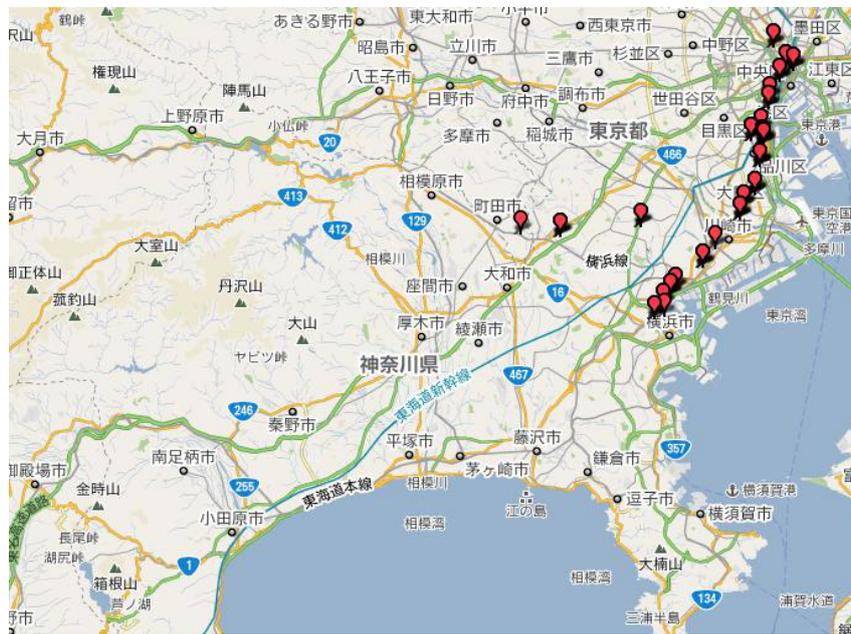


図 6.49: 10 区の時間帯に「駅伝」で検索した結果



図 6.50: 10 区コース

6.4 考察とまとめ

Twitter への投稿は現実世界の状況を反映しているということは、新幹線や山手線の例によって確認できる。また、AKB48 のライブの例は、ライブのように「その場所、その時間」だけの一過性のイベントも検出することができるということを示した。

箱根駅伝の検索結果はイベントを検出すると同時に、時間と状況によってイベントが移動することを検出できたと言える。箱根駅伝は 48 時間の内に東京箱根間の中で開催されるイベントであるが、盛り上がるポイントが常に移動するようなイベントを検出できるのは Twitter のようなマイクロブログとスマートフォンを組み合わせた利用形態がイベントに敏感に反応できるということを示していると言える。しかし、「箱根駅伝」という言葉を使用し、位置情報を付けて投稿していても選手が通り過ぎて時間が経った後に投稿されたものや全く関係のない文脈で使用されていると正確なイベント検知の妨げになる。今後の課題は、イベント検知のノイズとなる投稿の除去であると言える。

第7章 結論

本章では本研究のまとめと今後の課題について述べる。

7.1 本研究のまとめ

本研究は一般の人々にも手軽に位置情報が扱えるようになってきているという状況と、それを強力にサポートする様々な携帯端末やソフトウェアの登場に着目し、世間の状況に合わせて移り変わるイベントをマイクロブログを利用することによって検索できるのではないのかという動機から行われた。

ウェブログの登場によって人は世界に自由に自分のメッセージを発信できるようになり、パーソナルコンピュータが1台あってインターネットに繋がることさえできれば誰にも邪魔されずに自分の書きたいようにメッセージを書くことができる。さらに、Twitterを初めとするマイクロブログはスマートフォンとの親和性の良さから、人が表現する場所の制限を取り払った。携帯端末を片手に外に出て、写真やビデオを撮影し、位置情報を付けて「その場」でインターネットの世界に送り出すことができるようになった。この結果、位置情報を利用したサービスが興隆し、ますます位置情報の利用価値は増大している。

本研究では特にTwitterに付加された位置情報に着目し、投稿内容と共に分析することでキーワードと地理的位置を対応付けられるのではないかと考え、アルゴリズムの開発と実装を行った。その結果、「新幹線」というキーワードでは新幹線の路線の通りに綺麗にマップ上にピンが打たれ、「山手」というキーワードで検索すれば東京都内の地図に環状のピンが打たれた。また、2010年12月4日に東北新幹線の八戸、新青森駅間が開業し東北新幹線が全線開業となったが[34]、12月4日以降、新青森駅には「新幹線」というキーワードと共にピンが打たれ続けている。箱根駅伝の例は、24時間の中で激しく場所が移り変わっていくキーワードでもTwitterの位置情報とテキストがあれば検知できるということを示した。選手の往復と共にTwitterユーザの投稿も往復する様子は、現実世界の状況とインターネット上のテキストは関連しているということを示していると言える。

以上の事例が示すように、本研究の手法を用いれば携帯端末によって投稿されたマイクロブログのテキストを分析することで地理的位置とキーワードの対応付けが可能であると言える。

7.2 今後の課題と展望

本研究には位置情報と投稿内容のテキストが必要である。マイクロブログユーザが投稿しても位置情報を付加しなければシステムには使用できなくなってしまう。また、適切なキーワードを選ばなければ期待する結果が得られないこともわかった。箱根駅伝の例では、各大学の大学名で検索すれば各大学選手の走行を追うことができるのではないかと考えたが時系列に追うことができるだけのデータは得られなかった。本研究を行うにあたっては制限することなくTwitterのデータを収集している。このデータを利用すればイベント検知以外の研究にも応用することが考

えられる。データを継続して蓄積することで研究としてのより広い範囲で時系列の比較が可能になると期待できる。

参考文献

- [1] NTT レゾナント. goo 花火特集 2010 花火大会を地図から探す. <http://local.goo.ne.jp/season/hanabi/search/map.html?zm=5&lat=128470740&lon=502902170>. (document), 2.5
- [2] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010. (document), 3.1.1, 3.1
- [3] Twitter. 新年を新しい世界記録と共に迎えて. <http://blog.twitter.jp/2011/01/blog-post.html>. (document), 3.1.2, 3.2
- [4] 株式会社エヌ・ティ・ティ・ドコモ. キッズケータイ. http://www.nttdocomo.co.jp/product/foma/kids_phone/index.html. (document), 3.2.1, 3.4
- [5] Google. Google カレンダー. <https://www.google.com/calendar/>. 2.1
- [6] Google. Google マップ. <http://maps.google.co.jp/>. 2.1
- [7] 株式会社 角川マーケティング. Walkerpluse 花火カレンダー 2010 全国花火大会. <http://hanabi.walkerplus.com/calendar/>. 2.1.1
- [8] 東京都葛飾区. 第 44 回葛飾納涼花火大会. <http://www.city.katsushika.lg.jp/kurashi/213/021336.html>. 2.1.1
- [9] 隅田川花火大会実行委員会. 第 33 回隅田川花火大会. <http://sumidagawa-hanabi.com/>. 2.1.1
- [10] 総務省. 緊急通報の機能. http://www.soumu.go.jp/menu_seisaku/ictseisaku/net_anzen/hijyo/tuho.html. 2.2.2
- [11] 総務省. ブログ・sns の経済効果の推計. http://www.soumu.go.jp/main_content/000030547.pdf. 2.3
- [12] はてなダイアリー. ブログサービスごとの文字数制限の比較. <http://d.hatena.ne.jp/longlow/20080915/p1>. 2.3
- [13] Twitter Inc. Twitter. <http://twitter.com/about>. 2.3, 2.3.1
- [14] ITmedia.inc. Twitter と facebook, 自社公表された最新統計値のまとめ. <http://blogs.itmedia.co.jp/saito/2010/04/twitterfaaceboo.html>. 2.3.1
- [15] S. Milstein, A. Chowdhury, G. Hochmuth, B. Lorica, and R. Magoulas. Twitter and the micro-messaging revolution: Communication, connections, and immediacy 140 characters at a time, 2008. 2.3.1

- [16] 前嶋 和弘. オバマ政権とソーシャルメディア : 新しい政治コミュニケーションが動かす政策運営. 言語と文化, 22:99–114, 2010-3. 2.3.1
- [17] A. Burns and B. Eltham. Twitter free Iran: an evaluation of Twitter’s role in public diplomacy and information operations in Iran’s 2009 election crisis. 2009. 2.3.1
- [18] ITmedia. Twitter のジオタグ機能が twitter 本体でも可能に. <http://www.itmedia.co.jp/enterprise/articles/1003/12/news044.html>. 2.3.2
- [19] Fdration Internationale de Football Association (FIFA). 2010 fifa world cup south africa. <http://www.fifa.com/worldcup/archive/southafrica2010/index.html>. 3.1.2
- [20] 株式会社 毎日コミュニケーションズ. 日本代表決勝 t 進出の瞬間、twitter も史上最高の盛り上がり. 3.1.2
- [21] Apple inc. Apple. <http://www.apple.com/>. 3.2.1
- [22] Apple inc. iphone. <http://www.apple.com/iphone/>. 3.2.1
- [23] Apple inc. ios4. <http://www.apple.com/iphone/ios4/>. 3.2.1
- [24] 株式会社ナビタイムジャパン. ナビタイム. <http://corporate.navitime.co.jp/profile/index.html>. 3.2.1
- [25] 株式会社マピオン. ケータイ国盗り合戦. <http://kntr.jp/pc/>. 3.2.2
- [26] Inc. Foursquare Labs. foursquare. <http://foursquare.com/>. 3.2.2, 3.2.2
- [27] Mashable. Twitter and foursquare become the new loyalty program at tasti d-lite. <http://mashable.com/2010/01/13/tasti-d-lite-tastirewards/>. 3.2.2
- [28] 日経新聞社. 特集 1 2011 ヒット予測ランキング 1 位 得するジオゲーム. 日経 *TRENDY*, pages 20–21, 2010-12. 3.2.2
- [29] 株式会社ロケーションバリュー. イマナラ! <http://imanara.jp/index.html>. 3.2.3
- [30] 株式会社ロケーションバリュー. <http://www.locationvalue.com/>. 3.2.3
- [31] ITmedia. Twitter、ツイートのリアルタイムデータ「firehose」の提供先を拡大. <http://www.itmedia.co.jp/enterprise/articles/1003/02/news056.html>. 4.2
- [32] 株式会社 AKS. AKB48 公式サイト. <http://www.akb48.co.jp/index.html>. 6.3.1
- [33] Visit Zoo. AKB48 東京秋祭り. <http://www.visitzoo-akb48.jp/>. 6.3.1
- [34] 東日本旅客鉄道株式会社. 東北新幹線 新青森開業等について. <http://www.jreast.co.jp/press/2010/20100505.pdf>. 7.1

謝辞

本論文の作成ならびに研究活動のご指導を頂きました慶應義塾大学環境情報学部教授村井純博士、同学部教授中村修博士、同学部准教授楠本博之博士、同学部専任講師重近範行博士、同学部専任講師 Rodney D. Van Meter III 博士、同学部准教授三次仁博士、同学部教授武田圭史博士に感謝します。

また、常日頃から私を指導して下さった環境情報学部准教授植原啓介博士、政策・メディア研究科特別研究講師佐藤雅明博士には多大なる感謝をしております。特に、私は植原先生の授業を受け、その人柄に惹かれて村井研究室に入ることを決めました。先生との出会いを与えて下さった慶應義塾大学環境情報学部という場にも感謝しています。

iCAR グループの先輩であり、些末なことまで丁寧に指導して下さった波多野敏明氏、同期として切磋琢磨しあった鷓飼祐氏、澤田暖氏、Do Thi Thuy Van 氏、共に研究活動を楽しみ、努力した後輩の倉田彩子氏、村上滋希氏の皆様に感謝します。

私を関東の私立大学に通わせ、卒業するまで見守ってくれた両親には本当に感謝しています。二人の間に生まれて私は本当に幸せでした。

私の大学生活の大半を共に過ごした体育会競走部の仲間たち、競走部合宿所の寮生たちに感謝します。合宿所の寮母さんである神田艶子氏は私の関東での母親代わりのような存在でした。

体育会競走部の仲間として共に戦った主将の宮岸暖氏ほか、マネージャーパートの皆様、特に同期の田邊愛氏、森道代氏、古田あやか氏、山田はつみ氏の4人のマネージャーに感謝します。この4人がいなければ個性が強い部員が集まる競走部を円滑にマネジメントすることは不可能だったと思います。

卒業論文執筆にあたり、私を励まし支えてくれた競走部員の中島ミホ氏に感謝致します。

東京箱根間往復大学駅伝競走大会を共に運営した関東学生陸上競技連盟の仲間達に感謝致します。30数人という少ない人数であの規模の大会を運営するには、皆様との絆が無ければ成し得なかったと思います。

以上を以て謝辞とさせていただきます。ありがとうございました。