

# Place APIの統合

荒川豊<sup>1,a)</sup> Tatjana Scheffler<sup>2</sup> Stephan Baumann<sup>3</sup> Andreas Dengel<sup>3</sup>

**概要:** 近年、位置連携サービスの普及に伴い、POI (Point Of Interest) と呼ばれる、位置情報 (緯度・経度) に紐付けられた実世界に関する情報の重要性が高まっている。POI 情報は、Google や Facebook など様々なサービス提供会社が存在し、一般利用者に対しても、Place API という形で公開されている。しかしながら、それぞれ異なる ID や情報空間で管理していたり、各社の API の仕様や認証方式も様々であるため、利用者にとって利用しやすいとは言いがたい。そこで、本研究では、Foursquare, Facebook, Google, Twitter という 4 大サービスの Place API の仕様の差異を調査し、その差異を意識することなく、同一クエリーで問い合わせ可能にする代理アクセス機構の設計、開発を行う。さらに、同じ位置に対して戻されるレスポンスについても調査し、複数の Place API から得られた情報を統合する仕組みについて検討する。

**キーワード:** Point of Interest, 位置情報サービス, Place API

## Integration of Place API

**Abstract:** According to the recent spread of location-based services, the information about our real world associated with the coordinates, called POI (Point Of Interest), becomes much important. Major companies such as Google, Facebook have already started providing their POI data through API (Application Program Interface) which is generally called Place API for users. However, they manages POI by different ID and different accompanying information. Additionally, access ways such as an authentication are different with each other. In this paper, we investigate the difference of specification between Place APIs, and design and implement a proxy access interface which enables the developer to access 4 major Place APIs with the same query. Finally, we study how to integrate their different responses against the same coordinates.

**Keywords:** Point of Interest, Location-based service, Place API

### 1. はじめに

近年、Facebook や Twitter などに代表されるソーシャルネットワークサービス (SNS: Social Network Service) の普及が著しい。今や Twitter はユーザ数 5 億人を抱え、3 日で 10 億ツイートもの情報が発信されている。Facebook はユーザ数 10 億人に迫っており、1 日当たり 32 億コメント、3 億枚の写真投稿が行われている。そして、この膨大なデータから、新たなインテリジェンスを得ようとする研

究が盛んに行われている。

こうした膨大なデータの中でも位置情報が含まれるデータは、ソーシャルジオデータと呼ばれ、地震伝搬の分析 [2] や入力される日本語との相関関係分析 [1]、観光ルート分析 [3] など、さまざまな研究がなされており、我々も現在、各都市の人気観光スポットを自動的に抽出するシステムに関する研究に取り組んでいる。ソーシャルジオデータを観光に活用する研究は、[3] 以外にも [4][5] などがあり、これらの手法では、何らかのクラスタリング手法によって分析対象データをエリアに分割し、各エリア内におけるソーシャルジオデータの発生数をそのエリアの人気度と定義している。

ソーシャルジオデータの分析に用いられる、代表的なクラスタリング手法としては、Mean Shift Clustering[3][4] や p-DBSCAN[5] などがあるが、いずれのクラスタリング手

<sup>1</sup> 奈良先端科学技術大学院大学

Nara Institute of Science and Technology  
〒630-0192, 奈良県生駒市高山町 8916-5

<sup>2</sup> University of Potsdam

Karl-Liebknecht-Str. 24-25, 14476 Potsdam, Germany

<sup>3</sup> DFKI GmbH

Trippstadter Strasse 122, 67663 Kaiserslautern, Germany

a) ara@is.naist.jp

法でも得られる結果は、クラスタの中心座標とそのメンバ情報という数値データであり、このクラスタに対応する実世界の観光情報 (POI : Point of Interest) は不明である。分析結果を活用するためには、クラスタに対して何らかの実世界コンテキストを割り当てる必要がある。実世界コンテキストとは、建物などのオブジェクトだけではなく、開催されているイベントなども含んだ、人気の要因となるものである。このコンテキスト名は、従来研究において、ソーシャルジオデータ内のタグ情報から推定するもの [4] が一般的である。

それに対して、我々は、POI (Point Of Interest) と呼ばれる、位置情報 (緯度・経度) に紐付けられた実世界に関する情報データベースを用いた実世界コンテキストの推定 [6] を提案している。提案手法では、別のソーシャルジオデータ、具体的には、チェックイン\*1という行為に着目し、ソーシャルジオデータによるソーシャルジオデータへの自動ラベリングの達成を狙っている。これは、もともとスマートフォンのGPS精度が低いことから、チェックインサービスにおいて、スポット候補を提示する際に、距離だけではなく、人気も反映されているためである。距離に基づいた手法と比べて、誤差に対する耐性が強くなり、より確からしいラベルを得られるのではないかと考えている。その際、カテゴリによるフィルタリングを行うことによって、観光と無関係なスポット名を排除し、精度をさらに改善できると考えている。また、チェックインサービス事業者が保有する POI データベースには、正確性の高い POI 名が登録されていると期待できるため、従来のタグ分析による断片的な文字列よりも正確な名前を取得できると考えている。そして、文献 [6] では、Foursquare が提供している POI 情報を用いて、パリの人気観光スポット 3 地点の名前を推定したところ、Cathedrale Notre-Dame de Paris といった長い名前も正確に割り当てることが可能であることが示されている。

しかしながら、Foursquare に登録されていない POI は割り当てができない、あるいは Foursquare に登録されている POI 名が不正確である場合もある、といった問題がある。Foursquare は、ユーザーにチェックイン行動を促すため、ポイントによるゲーミフィケーションを取り入れている。その中で、新たに POI を登録する行為に対してポイントが付与されることになっており、ユーザーは積極的に新しい POI を登録する。その結果、POI データベースを Crowd Sourcing によって日々拡充できている。しかしながら、ユーザーによる登録は、不正確な POI 名や、情報の粒度が細かすぎる POI 名 (例えば、大きな駅ではホームごと、空港では搭乗口ごとに POI 情報が登録されていたりす

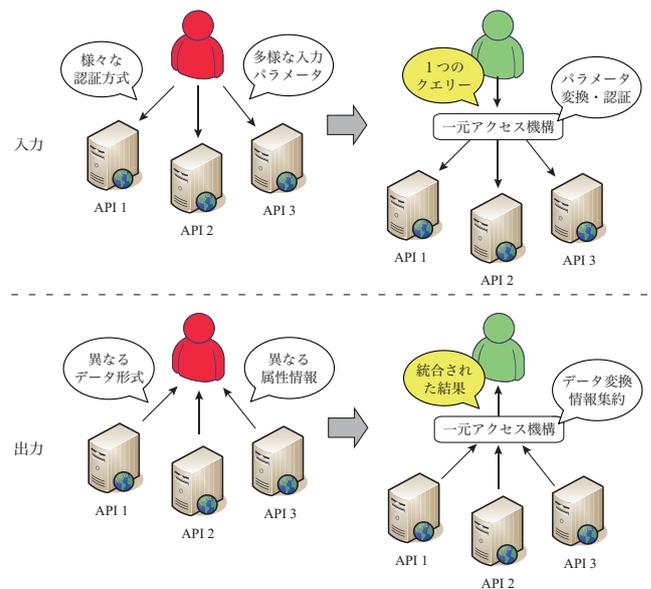


図 1 提案する一元アクセス機構の役割

る)の要因ともなる。

Foursquare の POI 情報は、Foursquare API\*2 という API (Application Programming Interface) を通じて、一般ユーザも利用することが可能であるが、こうした API は一般的に Place API と呼ばれ、Facebook (Facebook Places API\*3) や Google (Google Places API\*4), Twitter (Twitter Places API\*5), Yahoo (Yahoo GeoPlanet API\*6) など様々なインターネット企業が API を公開している。

そこで、我々は、Foursquare だけでなく、様々な Place API を複合的に用いることによって、それぞれの情報不足、情報ミスを補完し合うことができるのではないかと考えた。しかしながら、各 Place API は、認証方式、受け付ける入力データ、戻される出力データなど、形式的な仕様が異なっている。さらに、出力データの決定アルゴリズムは各社独自のものであり、同じ POI 情報の出力順位 (ランキング) も異なる。そのため、我々は Place API の統合に向け、1) Place API への一元的なアクセス機構の設計 (図 1 上) と、2) 出力データの統合手法 (図 1 下)、という 2 つの課題に取り組む。前者は、認証方式や受け付ける入力データの差異を吸収し、同一のクエリーで同時に複数の API へのアクセスを実現するものである。後者は、各 API から出力されたデータをマージし、同一 POI を表すものは含まれる内容を補完し合いつつ 1 つに集約する仕組みである。

本論文では、まず各社 Place API の仕様について調査し、その共通点と差異をまとめる。その上で研究課題 1) お

\*1 チェックインとは、Foursquare というサービスから始まったもので、ある場所にチェックインすることでバッジやポイントを獲得できるため、ユーザーが競ってチェックインをしている

\*2 <https://developer.foursquare.com/docs/>  
 \*3 <https://developers.facebook.com/docs/reference/api/>  
 \*4 <https://developers.google.com/places/documentation/>  
 \*5 <https://dev.twitter.com/docs/api/1.1>  
 \*6 <http://developer.yahoo.com/geo/geoplanet/>

表 1 Place API の比較

API 名	リリース日	チェックイン機能	カテゴリ指定	文字列指定	結果の並び順	位置の追加	認証
Foursquare API	Nov. 2009	○	○	○	Popularity	○	OAuth 2.0
Facebook Graph API	Aug. 2010	○	×	○	Popularity	×	OAuth 2.0
Google Places API	Nov. 2010	○	○	○	Prominence or Distance	×	API key
Twitter Places API	Jun. 2010	×	×	×	Unspecified	×	OAuth

表 2 Foursquare API に対する入力パラメータ一覧

変数	入力例	意味	備考
ll	44.3,37.2	中心座標 (緯度, 経度)	カンマ区切り, near かどちらかを指定
near	Chicago, IL	地名	Geocode Object*7 での指定も可
llAcc	10000	座標の精度 (m)	現在は検索結果に影響なし
alt	0	標高 (m)	現在は検索結果に影響なし
altAcc	10000	標高の精度 (m)	現在は検索結果に影響なし
query	donuts	検索語	POI 名に対する検索
limit	10	出力の上限 (件)	上限は 50 件
intent	checkin	アルゴリズム	checkin*8, browse*9, global*10, match*11の中から指定
radius	800	半径 (m)	上限は 10,000m
sw	44.3,37.2	南西の角の座標	領域を四角形で指定する場合
ne	44.1,37.4	北東の角の座標	領域を四角形で指定する場合
categoryId	asad13242l,btbe24353m	カテゴリ指定	カンマ区切り, Foursquare で定義された ID
url	http://nymag.com/	リンクする情報	サードパーティ向け
providerId	nymag	サードパーティの識別子	サードパーティ向け
linkedId	4247	サードパーティの ID	サードパーティ向け

よび 2) について説明し, 今後の課題について述べる.

## 2. Place API について

表 1 に, 有名な Place API について, チェックインサービスの有無やカテゴリフィルタが設定可能か否か, 結果の並び順などの観点からまとめる. チェックインサービスのパイオニアともいえる Foursquare が 2009 年 11 月に API をリリースしたのを皮切りに, メジャーなインターネット企業は次々にチェックインサービスを開始しており, その要素技術の 1 つである Place API の公開も始まっていることがわかる. 認証方式は, サービスの発展にともない変更されているが, 現在は OAuth と呼ばれる認証方式が主流となっている. 以降は, 入力と出力に分けて説明する.

### 2.1 入力パラメータの違い

まず例として, Foursquare API について説明する. Foursquare API では, ある位置に対する POI 情報を取得する場合, 表 2 に示す 15 個のパラメータを指定可能であ

る. 各パラメータは, 固有の変数名とそれに対するフォーマットが指定された入力値から構成されている. 例えば, 座標の指定は, ll=44.3,37.2 と「スペース無しカンマ区切りで緯度, 経度の順で ll という変数名」で API に渡す必要がある. また, POI 情報を取得する範囲は, 「座標 (ll) + 半径 (radius)」で円状に指定することもできれば, 「南西の角の座標 (sw) + 北東の角の座標 (ne)」で矩形で指定することも可能である. また, Foursquare API の場合, カテゴリを categoryId で指定することが可能であるが, この categoryId は Foursquare 社が独自に定義している識別子であり, 英数の組み合わせなので識別子からカテゴリ名を類推することはできない.

これらのパラメータのうち, 重要なものに関して, 各 Place API の仕様を表 3 に示す. 座標の指定に関しては, すべての API で変数名が異なる. さらに Twitter では, 緯度と経度を別の変数として渡す必要がある. 範囲に関しては, 変数と単位の違いがあり, 上限に関しては, パラメータとして存在しない API もある. カテゴリをパラメータとして設定可能な API は Foursquare と Google だけだが, 両者の変数名および複数のカテゴリを指定する際の区切り文字が異なる. さらに, Foursquare のカテゴリは, 9 つの主カテゴリと, その下に含まれる多数のサブカテゴリから構成される階層的なカテゴリとなっており, 主カテゴリを

\*7 Country Code や地名といった複数の情報を含んだオブジェクト

\*8 ユーザがチェックインしそうな POI を優先するモード

\*9 距離が近い POI を優先するモード

\*10 指定した位置を無視し, 検索語に基づいて全世界から抽出するモード

\*11 指定したエリアの中で, 検索語を含む POI を抽出するモード

表 3 入力パラメータに関する比較

変数	Foursquare	Facebook	Twitter	Google
座標	ll=44.3,37.2	center=44.3,37.2	lat=44.3&long=37.2	location=44.3,37.2
範囲	radius=100(m)	distance=100(m)	accuracy=5ft	radius=100(m)
上限	limit=10	limit=10	max_results=10	パラメータなし
カテゴリ	categoryId=c1,c2 階層のカテゴリ 例: 4bf58dd8d48988d181941735	パラメータなし	パラメータなし	types=c1 c2 フラットなカテゴリ 例: museum

表 4 出力データに関する比較

API 名	データ形式	POI の識別子	POI 名までのデータ構造
Foursquare	JSON か JSONP	Venue ID	response>venues>name
Facebook	JSON のみ	Numeric id	data>name
Twitter	JSON のみ	Geo ID	result>places>name
Google	JSON か XML	Place id	result> name

表 5 同位置に対する各 API からの上位 2 件の例

API 名	順位	POI 名
Foursquare	1	Cathédrale Notre-Dame de Paris
	2	Parvis Notre-Dame
Facebook	1	Cathédrale Notre-Dame de Paris
	2	Hotel Le Notre Dame
Twitter	1	Paris
	2	結果なし
Google	1	Square Jean XXiii
	2	Square René Viviani

指定することによって、下位のサブカテゴリすべてを指定することが可能となっている。一方、Google は、フラットな 126 のカテゴリから構成されている。また、その表記方法も異なり、Foursquare は長い英数字から構成された識別子であるが、Google は一般的な英単語である。そのため、表 3 の例で示すように、同じ美術館を指定する場合においても、大きく表記が異なる。

## 2.2 出力データの違い

まず、出力されるデータ形式の違いについて説明する。表 4 に示すように、Place API から出力されるデータ形式は、主に XML (eXtensible Markup Language), JSON (JavaScript Object Notation), JSONP (JSON with padding) の 3 つがある。表 4 に示すように、Foursquare は JSON か JSONP, Facebook と Twitter は JSON のみ、Google は JSON か XML, と若干異なる。また、その際に用いられる POI の識別子も各社で名称及び値が異なる。そして、同じ JSON 形式で取得したとしても、データ構造はすべて異なり、例えば、POI 名が記述された位置までのデータ構造も表 4 のような構造になっている。

次に出力される内容の違いについて説明する。すべての Place API は、ある位置に対して独自のアルゴリズムによって順位付けされたいくつかの POI 情報を出力する。その際、POI 名、緯度、経度、郵便番号、住所、電話番号、といった基本的な情報はどの API の結果にも含まれている。ただし、API によって、これらの情報が欠損している場合もあり、一円らが、ぐるなび、食べログ、Hotpepper というレストラン情報を提供する API に関して、レストラン情報を統合し、店の住所や電話番号を補完しあう研究を公表している [7]。基本的な情報以外にも、Foursquare は、カテゴリ、チェックイン数、like の数、dislike の数、メイ

ヤー\*12は誰か、誰が今ここにいるか、メニュー、価格、営業時間、Tips、写真、URL など、表 4 に示す 4 つの API の中で最も多くの情報を含んでいる。Google も同様に、基本情報以外にアイコンやユーザ評価、URL などの追加情報を含んでいる。一方、Twitter と Facebook は基本情報のみである。

同じ位置 (緯度 48.85316043, 経度 2.34935121) に対して得られる POI 情報は、各 API における選出アルゴリズムに依存しており、表 5 に示すように大きく異なる結果となる。この例の場合、Cathédrale Notre-Dame de Paris という POI が Foursquare と Facebook の両 API から取得できるため、この位置に対する POI は Cathédrale Notre-Dame de Paris ではないかと推測できる。しかしながら、あくまで推測であり、この結果を如何に統合するかは重要な課題となる。

## 3. Place API の統合に向けて

Place API の統合は、序論に述べたように、1) さまざまな Place API への一元的なアクセス (入力) と、2) Place API から得られる出力の統合 (出力) という 2 つに大別できる。さらに後者は、一円ら [7] や Tatjana ら [8] のように、同一 POI と思われる情報を 1 つに統合する (個別 POI 情報の統合) と、指定した位置に対して得られる複数 API からの複数 POI 情報を確からしい順に順位付けし統合する (POI ランキング) の 2 段階に分けることができる。

\*12 メイヤーとは、Foursquare において、ある POI に対して最もチェックイン (来訪歴) が多い人に与えられる称号

表 6 一元アクセス機構への入力パラメータ

意味	変数名	備考
ユーザ Key	key	認証用, 必須パラメータ*13
緯度	lat	必須パラメータ
経度	lon	必須パラメータ
範囲	rad	単位は m で指定. 未指定の場合は 100m
上限	max	未指定の場合は 10 件
文字列	txt	最大文字数 10 文字
カテゴリ	typ	Google の ID, Foursquare の ID 独自の ID 区切り文字は : (コロン)
API 指定	prv	F: Foursquare, G: Google, T: Twitter f: Facebook, 例: FfG

表 7 抽象度の高い独自カテゴリの設定の例

独自 ID	対応するカテゴリ ID	
	Foursquare (内は意味)	Google
travel	4fceea171983d5d06c3e9823(Aquarium)	aquarium
	4d4b7104d754a06370d81259(art & entertainment)	museum
	4deefb944765f83613cdba6e(Historic Site)	park
	4bf58dd8d48988d181941735(Museum)	city hall
	4bf58dd8d48988d182941735(Theme park)	zoo
	4bf58dd8d48988d1d941735(Bridge)	art gallery
	4bf58dd8d48988d163941735(Park)	neighborhood
	4bf58dd8d48988d161941735(Lake)	place of worship
	4bf58dd8d48988d129941735(city hall )	establishment
	4bf58dd8d48988d1f9931735(road)	sublocality
	4bf58dd8d48988d12d941735(monument landmark)	political
	4bf58dd8d48988d17b941735(Zoo)	church
	4bf58dd8d48988d164941735(Plaza)	
	4d95416a243a5684b65b473(Rest Area)	
	4fbc1be21983f883593e321(well)	
	4bf58dd8d48988d129951735(trainstation)	
	5032792091d4c4b30a586d5c(Concert hall)	
	4bf58dd8d48988d15a941735(Garden)	
	4bf58dd8d48988d184941735(stadium)	
	4bf58dd8d48988d132941735(church)	
4bf58dd8d48988d1f2931735(performing Arts venue)		
4bf58dd8d48988d1fa941735(Farmers Market)		

### 3.1 入力インターフェースの統合

システムの構成は、図 1 右上に示すように、各 Place API に対して代理アクセスを行う一元アクセス機構を設置する。ユーザに対する共通のクエリは、各 API の共通点を洗い出し、表 6 に示す 7 つのパラメータに制限する。

一元アクセスを実現可能にするために入力パラメータを制限する弊害として、各 API 特有の機能が使えなくなるが、今回は利便性を重視しており、問題ではないと考えている。認証用のユーザ Key と、位置を指定する緯度と経度のみを必須パラメータとし、それ以外のパラメータはオプション扱いとする。ユーザ Key とは、一元アクセス機構へのユーザ登録時に生成されるキーであり一元アクセス機構に対する認証に利用される。各 API の OAuth 認証に必要な API key や User secret といった情報は、一元アクセス機構にユーザ登録する際に登録しておくことで、実際に API にアクセスする際の入力を不要としている。また、取得範囲や上限の件数などは、各 API における設定可能な上限、下限にあわせて変更される。

カテゴリについては、Foursquare と Google を指定した場合にのみ有効になる。Foursquare と Google ではそれぞれ ID 空間が異なる上、特に Foursquare は細分化されていることから、我々は独自にこれらを分類しなおし、新しいカテゴリ ID 空間も作成する。独自のカテゴリでは、表 7

に示すように、「travel」といった抽象度の高いカテゴリを設定し、それぞれに対して、Foursquare および Google のカテゴリ ID を割り当てる。抽象度の高いカテゴリとすることにより、ユーザの利便性の改善を狙っているが、現在は、手動で分類しており、その割当が適当ではない可能性もある。今後、ユーザ独自のカテゴリセットの作成機能を実装し、多ユーザが設定した組み合わせ情報から、自動的にカテゴリ振り分けを行う仕組みや、オントロジーによるカテゴリの分類、嗜好に基づいたカテゴリの分類などを検討していく予定である。なお、変数の指定は、カテゴリ ID をコロン区切りで記述し、複数の ID 空間を混在して指定することも可能とする。

### 3.2 出力データの統合

出力データの統合は、個別 POI 情報の統合と、POI ランキングの 2 段階に分けることができる。今回、利用する 4 つの API の中では、ユーザによる POI の登録機能を持つ Foursquare が質、量ともに多いと考えられる。そこで、Foursquare の結果をキーとして、そこで得られた POI 名を txt パラメータとして別の API にアクセスし、同一と思われる情報があれば統合する。一円ら [7] は、住所を正規化する\*13 API を利用した上で、類似住所のデータをグループ化し、その中で、電話番号と店舗名による同一判定を行なっている。この手法は、日本の住所表記ルールの場合、正規化 API も存在することもあって、動作可能であるが、海外の住所に対しては利用できないという問題がある。そのため精度は低下すると考えられるが、我々は名前による比較によってのみ同一判定を行う。文字列の類似度計算には、メジャーな手法として、Levenshtein 距離と Jaro-Winkler 距離がある。Levenshtein 距離は、文字列全体を対象として、その編集距離をスコアにしたものであり、数字が小さいほど類似度が高いことを示す。一方、Jaro-Winkler 距離は、ある範囲の中で交換可能な文字がないかを探索するアルゴリズムであり、軽微なスペルミスなどにはこちらが適していると考えられるため、提案システムでは、Jaro-Winkler 距離を採用する。現システムでは、2 つの POI 名の Jaro-Winkler 距離が 0.8 以上である場合、その POI は同一であると判断している。このしきい値は、ヒューリスティックに決定したものであり、今後、さまざまな POI に関して、定量的評価を行い、適切なしきい値を決定する必要がある。このようにして統合されたそれぞれの POI 情報に対して、A) 各 Place API からの出力のうち上位の POI の方が有名、B) 複数の Place API から得られた POI ほど有名、C) Foursquare におけるチェックイン数が多い POI ほど有名、という 3 つの前提に基づき、各 POI のスコアリングを行うことにより、得られたすべての

\*13 住所の正規化とは、「1 丁目 2 番地」と「1-2」といった表記揺らぎを一定のルールに統一すること

```
object {3}
  param {7}
    lat : 43.234
    lon : 134.564
    max : 10
    key : sample
    rad : 100
    typ : travel
    prv : FfG
  results : 3
  poi [3]
    0 {5}
      name : POI名
      api : Foursquare
      rank : 1
      id : 各APIで使用されているID
    merged [2]
      0 {4}
        name : POI名
        api : Facebook
        rank : 2
        id : 各APIで使用されているID
      1 {4}
    1 {5}
    2 {5}
```

図 2 出力データの例

POI を順位付けする。

図 2 に提案システムから出力される結果の例を示す。出力形式は、JSON 形式として、入力されたパラメータ (param)、得られた結果の数 (result)、そして、各 POI 情報から構成される。各 POI 情報は、提案アルゴリズムによって順位づけされた順に出力されるものとし、それぞれの POI 情報には、POI 名、そのソース API、ソース API 内での順位、ソース API 内での識別子が含まれている。さらに、同一と判定されて統合された他のソース API からの結果も含まれる。

これらの出力形式は、現在、我々が評価用アプリケーションとして構築中の観光案内に必要な情報だけに絞り込んであるが、今後、提案システムの応用先に応じて、出力される情報、形式を再検討する必要があると考えている。

## 4. おわりに

本論文では、さまざまな会社によって提供される Place API の差異をまとめ、その差を意識することなく利用するための一元アクセス機構を提案した。提案は、各 API で共通となるパラメータに限定した入力パラメータとすることにより、すべての API に対して同一のクエリーによりア

クセスを可能にしている。さらに、出力結果は、同一 POI を名前の類似度から統合するとともに、その出現頻度やチェックイン情報に基づいて独自の重み付けを行い、指定された位置における知名度の順で出力する。今後、提案システムを、Github などに公開し、オープン化すると共に、出力結果の統合に関して、定量的な評価を行なっていく予定である。

**謝辞** 本研究の一部は、財団法人人工知能研究振興財団の研究助成に基づくものである。ここに記して謝意を示す。

## 参考文献

- [1] 荒川豊, 田頭茂明, 福田晃, “[推薦論文]Twitter を用いたコンテキストと入力文字列の相関関係分析,” 情報処理学会論文誌, Vol.52, No.7, pp.2268–2276, 2011 年 7 月.
- [2] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors,” In Proc. of the 19th International Conference on World Wide Web (WWW), pp. 851–860, 2010.
- [3] D.J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, “Mapping the world’s photos,” In Proceedings of the 18th international conference on World wide web, pp. 761–770. ACM, 2009.
- [4] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, “Travel route recommendation using geotags in photo sharing sites,” In Proc. of the 19th ACM international conference on Information and knowledge management, pp. 579–588. ACM, 2010.
- [5] S. Kisilevich, F. Mansmann, and D. Keim, “P-dbscan: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos,” In Proc. of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, p. 38. ACM, 2010.
- [6] 荒川豊, 福田晃, “ソーシャルジオデータのクラスタリング結果に対する自動的な意味付けに関する一検討,” 第 75 回情報処理学会全国大会, No.6D–4, Vol.3, pp.7–8, 2013 年 3 月.
- [7] 一円真治, 梶克彦, 河口信夫, “ハイブリッド位置情報サービスのための POI 情報の統合プラットフォーム,” 第 75 回情報処理学会全国大会, No.6V–9, Vol.3, pp.193–194, 2013 年 3 月.
- [8] Tatjana Scheffler, Rafael Schirru, and Paul Lehmann, “Matching Points of Interest from Different Social Networking Sites,” KI 2012: Advances in Artificial Intelligence - 35th Annual German Conference on AI, pp.245–248, 2012.