

POI 情報統合プラットフォームの提案

一円 真治¹ 梶 克彦¹ 河口 信夫¹

概要：スマートフォンといった GPS 機能を搭載したデバイスの普及により、ユーザの位置情報に応じて、飲食店やホテルなどの POI(Point Of Interest) の情報を提供する位置情報サービスが多数提供されている。位置情報サービスには、ユーザが欲しい情報を個別のサービスからは網羅的に取得できない、取得できる POI の属性情報が各サービスによって異なるなどの未解決な課題が存在する。また、ユーザが屋内に位置する場合、自動販売機やロッカーなどの POI をフロアマップや Web ページから網羅的に知ることができない問題がある。本研究では以上の課題、問題点を解決するために、POI の情報統合を行うプラットフォームを提案する。統合情報は Linked Data として利用するために、RDF で出力可能とする。また、同一の POI に関して情報統合を行うために、POI の住所・電話番号・名前情報を用いた同定手法を提案する。提案する同定手法には、POI 検索判定と名前文字列判定があり、判定する POI に応じて 2 つの同定手法を使い分けることが効率的なことを確認するため、評価実験を行った。結果、適合率：1.0、再現率：0.91 であった。

Proposal of a Platform Integrating POI Information

SHINJI ICHIEN¹ KATSUHIKO KAJI¹ NOBUO KAWAGUCHI¹

1. はじめに

現在、スマートフォンといった GPS 機能を搭載したデバイスの普及により、ユーザの位置情報の取得が容易になってきている。これに伴い、ユーザの現在位置に応じて適切な情報提供を行う位置情報サービスが多く登場している。位置情報サービスの例として、Foursquare[1] のチェックイン、Google Maps[2] の地図表示や目的地へのナビゲーション、Hot Pepper[3] の周辺の飲食店検索がある。位置情報サービスの内で、飲食店やホテルなど POI(Point Of Interest) の情報提供を行うサービスが多くて提供されており、POI に関して注目が集まっていることがわかる。POI に関して提供する情報を各サービスが特徴付けることで、種類が多くなり多様化している現状がある。また、位置情報サービスの多くは Web API を公開している。これを用いることでシステム開発者がサービスが保持しているコンテンツの利用して、Web サービスを開発することが可能となっている。しかし、POI に関する位置情報サービスには

課題がある。我々の調査の結果、以下の課題が解決されていない [4]。

- (1) ユーザが欲しい情報を個別の位置情報サービスからは網羅的に取得できない
- (2) 各位置情報サービスが持つ属性情報が異なる
- (3) 位置情報サービスにより POI の属性情報の記述フォーマットが異なる
- (4) 位置情報サービスから取得できる情報に誤った情報が含まれている場合がある

また、ユーザが商業施設や地下などの屋内に位置する時に、位置するフロア内の POI の情報を網羅的に知りたいならば、その商業施設や地下のフロアマップや Web サイトを見て調べると考えられる。しかし、実際にはそれらに自販機やロッカー、トイレなどの小規模な POI は掲載されていない場合があり、必要な時にそれらの情報を知ることができない問題がある。

屋外・屋内問わずユーザが網羅的に POI を得ることを可能とするために、本研究では複数の位置情報サービスから得られる情報、ユーザによる POI に対する生成情報など、POI に関連性のある情報の統合を行うプラットフォームを

¹ 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

提案する。POIに関する情報統合を行うためには、名寄せを行う必要がある。名寄せを行うために、同一地点のPOIが取得した情報内に複数含まれているかどうか判断する同定手法を提案する。提案する同定手法では、POIの住所・電話番号・名前情報を用いて同定を行う。

本プラットフォームでは、ユーザからのクエリを緯度・経度・検索半径などから構成し、それに基づいて複数の位置情報サービスから取得した情報を本プラットフォームのPOI情報統合部分が情報統合を行い、より網羅性の高い情報をユーザに返す。本稿では、2章で、POIについて述べ、3章で、POI情報統合プラットフォームの提案を行う。また、4章では、統合手法について、5章で評価、6章でまとめと今後の課題について述べる。

2. POI(Point Of Interest)

POIとは、Point Of Interestの略称であり、誰かが興味があると思った特定の場所を指す。具体例としては、飲食店や観光名所、ホテルなどがある。屋内にあるようなロッカーやAED、自動販売機などの小規模な物もPOIである。本稿では、名前、緯度経度、住所などの属性情報をすべて含めた情報をPOIとして定義する。POIは主に地物情報を指すことが多いが、以前に閉店した店舗やこれから開店する店舗、ある場所で一時的に催されるイベントなど時間情報を考慮した物もPOIとして考える。以下より、POIを取得できる位置情報サービスと屋内のPOIについての調査を述べる。

2.1 位置情報サービス

POIを取得できるWeb APIを多くの位置情報サービスが公開している。例えば、Foursquare, Google Places[5], Hot Pepper, Yahoo!ローカルサーチ API[6], ぐるなび [7], 食べログ [8]などが挙げられる。Web APIはHTTPを用いており、各位置情報サービスが保持するコンテンツデータをユーザは利用することが可能である。各位置情報サービスのWeb APIが定めたURLのパラメータに取得したい情報の値を設定することで、取得できる情報を選択することができる。取得データの多くはJSON, XML形式であり、使用する位置情報サービスによって選択可能であるかは決まっている。またWeb APIを利用するには、サービスごとのIDを取得する必要がある。また各位置情報サービスが定めるライセンスを守る必要がある。ライセンスの例としては、1日あたりのリクエスト回数や取得したデータの扱いなどがある。

2.1.1 調査

POIを取得できるWeb APIを公開しているHot Pepper, ぐるなび, 食べログの3つの位置情報サービスから名古屋駅を中心とした半径1.0km(食べログはWeb APIの仕様上1.5km)の範囲内に含まれる飲食店のPOIに関する情報

を取得しデータの分析を行った(調査日:2013年1月)。取得できたPOI数はHot Pepper:569件, ぐるなび:484件, 食べログ:1021件であった。POIの重複の調査の結果を図1に示す。Hot Pepper, ぐるなびとは、飲食店側が情報を提供する運営形式であるので、POIの重複が多いことが確認できた。一方、食べログの飲食店情報は食べログに登録しているユーザが投稿したものであるため、Hot Pepperとぐるなびからは得られないPOIが多く取得できることが確認できた。以上のことから、各位置情報サービスから取得できるPOIにばらつきがあることが確認できた。

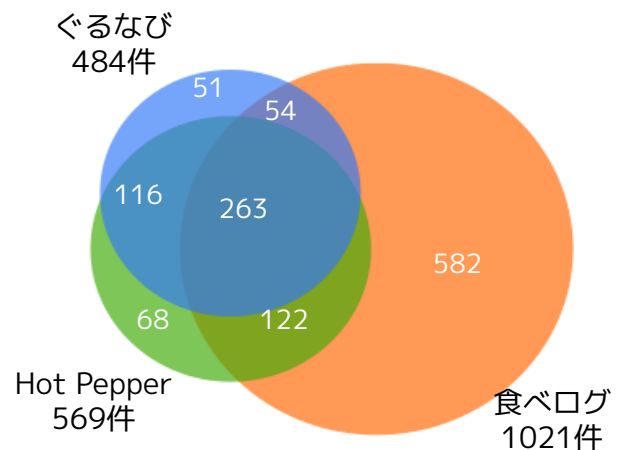


図1 POIに対して重複の有無の調査

1つのPOIに関して取得できる属性情報についても調査を行った。POIの属性情報をGoogle Places, Hot Pepper, ぐるなび, の各位置情報サービスから取得し、取得できた属性情報の例とその違いを表1にまとめた。ぐるなびから取得できた名前情報には(赤から鍋とせせり焼き)という店舗のキャッチコピーが含まれている。住所情報では県名や丁番号の表記に違いがあり、建物名やフロア名の情報は個々の属性情報ではなく、住所情報の一部として取得され、建物名には略称が用いられる場合があることが確認できた。Hot Pepperからは電話番号情報を取得できず、Google Placesでは国際番号表記であった。飲食店情報を提供するHot Pepperとぐるなびの2つのサービスにおいても取得できる属性情報に違いがあることも確認できた。これらより、POIに関して取得できる属性情報とそのフォーマットは各サービスによって異なることが確認できた。

表 1 POI の属性情報 [赤から名駅店]

属性情報	位置情報サービス名		
	ぐるなび	Hot Pepper	Google Places
名前	赤から鍋とせせり焼き 赤から 名駅店	赤から 名駅店	赤から 名駅店
住所	愛知県 名古屋市 中村区名駅 3-14-16 東洋ビルディング B1	名古屋市 中村区名駅 3-14-16 東洋ビル B101	愛知県 名古屋市 中村区名駅 3 丁目 14-16 東洋ビルディング B1F
緯度	35.172474	35.172452	35.172220
経度	136.883723	136.8838750	136.8843340
電話番号	052-588-xxxx	なし	+81 52-588-xxxx
営業時間	○	○	○
ジャンル	×	○	○
アクセス	○	○	×
最寄り駅	○	○	×
収容人数	×	○	×
喫煙	×	○	×
口コミ	×	×	×
PR 文	○	○	×
クーポン	○	○	×

2.2 屋内の POI の調査

建物の各フロアや地下などの屋内には、多くの POI が存在していると考えられる。一般的に屋外などであれば Google Maps や Yahoo!地図 [9] などの地図アプリケーションで POI を調べることは可能であるが、屋内の POI を知りたい場合に関しては一般的に建物のフロアマップやサイトが Web で公開されていれば、それらからいつでも閲覧可能である。しかし、未公開ならば、建物内に設置されているフロアマップを見なければならぬ。Google Maps や Yahoo!地図などの地図アプリケーションは多くの場合、屋外の POI を調べる際に活用するが、共に一部地域や施設に関しては屋内地図を閲覧できる機能を提供している。屋内地図には店舗、トイレ、ATM、出入口などの POI は表示されている。しかし、実際には自動販売機や AED、ロッカー、公衆電話など多くの屋内における POI が屋内地図に表示されていない場合があり、ユーザはこれらの情報は現地に行かないと知ることができないという問題点がある。また、屋内地図の POI は種類であったり、名前が表示されているだけで、例えば洋式のトイレなのか、どこの金融機関の ATM であるのかなどの詳細な属性情報は屋内地図からは知ることができない。

名古屋駅地下 1 階のテルミナ・メイチカ、ミッドランドスクエア地下 1 階のエリアに関して、Yahoo!地図の屋内地図に表示されていない POI に関する調査を行った (調査日: 2013 年 4 月)。結果を表 2 に示す。屋内には、自動販売機や ATM、公衆電話といった設置や撤去される可能性が高い POI が多く存在するため、屋内地図の管理は容易ではないと考えられる。

表 2 屋内地図に表記されていない POI

POI の種類	個数
自動販売機	6
公衆電話	5
ロッカー	4
ATM	4
AED	2
消火器・非常ボタン	6

3. POI 情報統合プラットフォーム

提案するプラットフォームの概要を図 2 を示す。本プラットフォームを構成する要素として、POI を利用したいクライアントと POI の情報の統合や管理を行うシステム、情報の取得元である位置情報サービス、Web ページ、ボランティアを活用したクラウドソーシングがある。クライアントとは、一般ユーザと開発者を指す。システムはクライアントと複数の位置情報サービスとのハブの役割を果たし、クライアントは 1 つのクエリで複数のサービスにリクエストを送ることができる。クライアントが緯度・経度・検索半径・キーワードなどを元に作成したクエリをシステム部のクライアント管理部が受け取り、各サービス管理部がそれに対応したリクエストに変換し、情報のやり取りを行う。サービス管理部は単一のサービスのみを管理しており、この管理部をプラグインという形でシステムに対し削除・拡張可能とし、クライアントが利用したい位置情報サービスを選択可能とする。

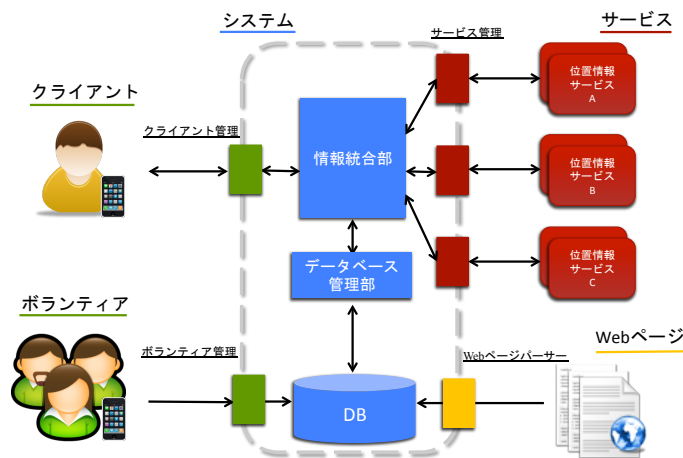


図 2 POI 情報統合プラットフォーム

3.1 プラットフォームの動作

クライアントが本プラットフォームを利用した場合、取得できる POI の情報がデータベースに保存されているかないかで、システムの動作は異なる。以下で動作について述べる。

複数の位置情報サービスから取得した POI をシステム

の情報統合部が受け取る。次に、取得できた POI の属性情報の正規化を行い、フォーマットを整える。これにより、1 章で述べた位置情報サービスの課題 (3) を解決する。そして、フォーマットが整えられた POI を統合する。複数の位置情報サービスから取得した POI を統合し、単一の位置情報サービスから取得できる情報よりも網羅性を高めることで、課題 (1) を解決する。取得した情報の中に同一の POI が含まれていれば、1 つの POI に関して重複していない属性情報を追加して情報量を拡張し、属性情報に関して、より網羅性の高い POI の統合情報をクライアントに返す。このことにより、課題 (2) を解決する。

データベースに情報が保存されている場合は、クライアントからのクエリをデータベースにも送り、クエリの条件を満たす POI を取得し、位置情報サービスから得られた POI と比較する。互いの取得した情報で不足している情報は補い合い、データベースからの情報に修正情報が含まれていれば位置情報サービスから取得した情報に修正を加える。そして、以上の過程を経た POI の統合情報はシステム部のデータベース管理部によりフォーマットが整えられ、データベースの情報との差分情報が保存される。情報統合のための POI 同定手法に関しては、次章で述べる。クライアント管理部がシステムとの繋がりを管理しており、クライアントの需要に合った利用を可能とする。

3.2 POI の属性情報の取得

位置情報サービスの Web API から取得できる POI の属性情報は、各位置情報サービスが提供している属性情報しか取得できない。建物名やフロア名、口コミなどの属性情報を取得する方法を以下で述べる。

3.2.1 Web ページ

デパートや駅などの大きなエリアを持つ POI は公式の Web ページを持っている場合が多い。公式 Web ページがある POI に関しては、その Web ページの HTML をパースすることで得たい POI の属性情報を抜き出す。抜き出す情報としては、建物名やフロア名を考えている。なぜなら、それらは位置情報サービスによっては略称が用いられているため表記が統一でない場合や、住所情報や名前情報に含まれている場合があり、それらの内で、正確な建物名がどれであるのかの判断が困難であると考えからである。Web ページの情報を直接入力可能とする編集ツールでデータ編集の際の負担なくし、各 Web ページ用のパーサーを作成することで属性情報の取得を自動化する。

3.2.2 クラウドソーシング

Web 上や地図にない屋内外の自動販売機や公衆トイレなどの POI 自体の情報、また、その自動販売機では何が売られているか、そのトイレは清潔に保たれているかどうかなどの POI の属性情報は、多種多様に考えられ、個人がすべての情報を収集や追加、編集を行うことは、不可能で

ある。そこで、クラウドソーシングを活用し、POI の収集や属性情報の追加、誤った情報の編集を行う。取得できる POI には誤った情報が含まれていることがあるという課題 (4) はクラウドソーシングにより解決する。また、クラウドソーシングは情報の収集や編集などに広く活用されている [10], [11]。先に述べた Web ページのパーサー作成や、Web ページからの情報の直接入力もクラウドソーシングを活用する。

3.3 POI の統合情報の保存

同一と判定された POI の統合情報は、システムのデータベースに保存する。データベースからデータを取得する際は、RDF 形式で出力する。これは、POI の統合情報を Linked Data として利用することを想定しているからである [12], [13]。

3.3.1 統合情報の構造

統合情報は RDF 形式で出力可能とする。RDF はデータの関係を主語、述語、目的語という 3 つの要素 (トリプル) で表現する。データ構造を図 3 に示す。図 3 において、主語は POI ID、述語は各矢印の上部に表記し、目的語は各矢印の先のボックスで表現している。POI ID は POI の統合情報に独自に割り当てた ID である。また、述語の定義には、Dublin Core や Geo Vocabulary などの共通のメタデータ記述に使用する語彙を活用する。

多くの位置情報サービスには、Web API から取得した情報を長期間保持してはならないというライセンスの問題があるため、すべての情報を保存せず、公開情報かつ、ライセンスを満たすと考えられる属性情報を保存する。保存する情報は、1 つの POI に対して各位置情報サービスが割り当てている ID とライセンス管理のための取得した日付、名前、緯度経度、住所、建物名、フロア名、電話番号、カテゴリーである。各サービスの ID を保持することで POI と他の位置情報サービスを紐付ける。なお、図 3 における service_ID と category はトリプルを POI ID に関連付けることで拡張可能である。また、位置情報サービスや Web ページから取得できないような詳細な情報も同様に追加することが可能である。自動販売機や AED などの屋内の POI は、緯度経度と属する建物、フロア名、カテゴリーを関連付けることで表現可能である。

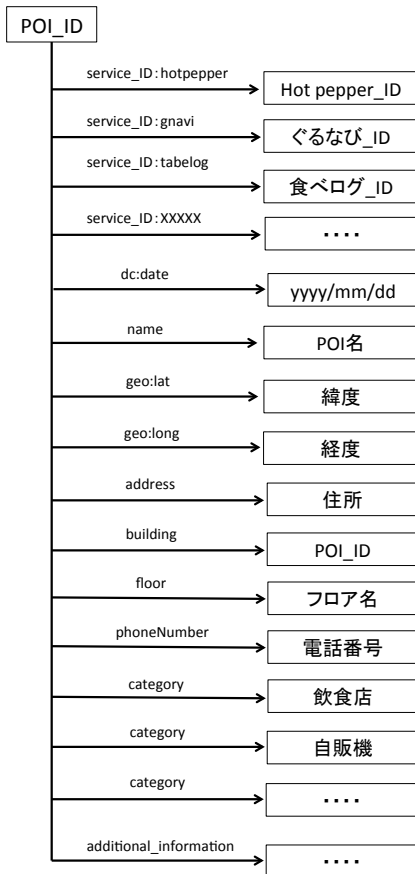


図 3 統合情報のデータ構造

```

{
  "name": "まるや本店 名駅店",
  "address": "愛知県名古屋市中村区名駅1-2-1 名鉄百貨店本館9F レストラン街",
  "latitude": 35.16862977874623,
  "longitude": 136.885028933887,
  "phoneNumber": "052-585-7108",
  "hotpepperSource": 1,
  "tabelogSource": 1,
  "gnaviSource": 1,
  "gnaviPOIObject": {
    "update_date": "2013-02-03 05:51:33",
    "name_kana": "ひつまぶし・会席",
    "url": "http://r.gnavi.co.jp/n658401/7ak/u003dgyhq9v0jultvu49d0ak9OeGG8eTV6mtLhK2F1qhkCUukK3D",
    "url_mobile": "http://mobile.gnavi.co.jp/shop/n658401/7ak/u003dgyhq9v0jultvu49d0ak9OeGG8eTV6mtLhK2F1qhkCUukK3D",
    "shop_image1": "http://apicache.gnavi.co.jp/image/rest/index.php?img/u003dn658401w.jpg/u0026sid/u003dn658401",
    "shop_image2": "",
    "qrCode": "http://apicache.gnavi.co.jp/image/rest/index.php?img/u003dn658401qr.png/u0026sid/u003dn658401",
    "tel": "052-585-7108",
    "fax": "052-585-7108",
    "openTime": "11:00~23:00(L.O.22:00)",
    "line": "名鉄名古屋本線",
    "station": "名鉄名古屋駅",
    "walk": "5",
    "note": "",
    "pr_short": "本物の味！名古屋名物ひつまぶし専門店。食べて納め、行列覚悟でご来店下さい。",
    "pr_long": "名鉄駅の名鉄百貨店9階にあるご存知名古屋名物「ひつまぶし」のお店。線はその時期に一審限ののったものを厳選し、ひつまぶしの命でもあるたの味を是非一度ご堪能下さい。",
    "areaCode": "AREA130",
    "areaName": "中部",
    "areaName_s": "名駅前・国際センター",
    "category_code_l_0": "CTG100",
    "category_name_l_0": "和食",
    "category_name_s_0": "和食",
    "category_code_s_0": "CTG117",
    "category_name_s_1": "和食",
    "category_code_s_1": "CTG102",
    "equipment": "貸切可,店内全面禁煙,ソフバンク,NTT ドコモ,au,法事で利用可,結婚で利用可,",
    "budget": "2500",
    "mobile_site": "1",
    "pc_coupon": "1",
    "id": "n658401",
    "source": "GnaviRestSearch",
    "name": "まるや本店 名駅店",
    "latitude": 35.166447,
    "longitude": 136.886989,
    "address": "〒450-0002 愛知県名古屋市中村区名駅1-2-1 名鉄百貨店本館9F レストラン街",
    "phoneNumber": "052-585-7108",
    "normalizedAddressObject": {},
    "addressElement": {
      "prefecture": "愛知県",
      "extraLargeArea": "名古屋市",
      "largeArea": "中村区",
      "mediumArea": "名駅",
      "smallArea": "1-2-1",
      "building": "名鉄百貨店本館9F レストラン街"
    }
  },
  "hotpepperPOIObject": {
    "logo_image": "http://imgfp.hotpepper.jp/IMGH/31/26/P001073126/P001073126_69.jpg",
    "name_kana": "まるや本店 名駅店",
    "station_name": "名古屋",
    "ktal_coupon": "1",
    "largeServiceArea": {
      "code": "S530",
      "name": "東海"
    },
    "serviceArea": {
      "code": "SA33",
      "name": "愛知"
    }
  }
}

```

図 4 JSON 形式の統合情報の一部

3.4 Web アプリケーションの実装

提案するプラットフォームの実装として、Web アプリのプロトタイプシステムを開発した。本節では、プロトタイプシステムの概要について述べる。

3.5 プラットフォームの実装

提案するプラットフォームにおいて、システムの情報統合部を Java で実装した。ユーザのクエリとして緯度・経度・半径を入力すると、その値に基づいて各位置情報サービスから POI を JSON 形式で取得する。JSON 形式のデータをすべて Java オブジェクトに変換し、次章で詳しく述べる統合手法により同一の POI ごとに 1 次元リストを作成する。同一の POI の 1 次元リストを複数持つ 2 次元リストを最終結果とし、JSON 形式に変換し、出力する。図 4 に出力された JSON 形式の統合情報の一部を示す。

3.6 クライアントアプリケーションの実装

Play!Framework を用いて実装を行った。図 5 にプロトタイプシステムをブラウザで起動した初期画面を示す。この画面では、クライアントが現在位置をキーワードか緯度経度で入力し、POI の検索半径を 100m, 600m, 1000m から決定する。現在位置をキーワードで入力した場合は、

Google Geocoding API[14] を用いてジオコーディングを行うことで、緯度経度を取得している。なお、今後はクライアントが現在位置を自発的に入力するのではなくブラウザや GPS から自動的取得できるように改良する予定である。

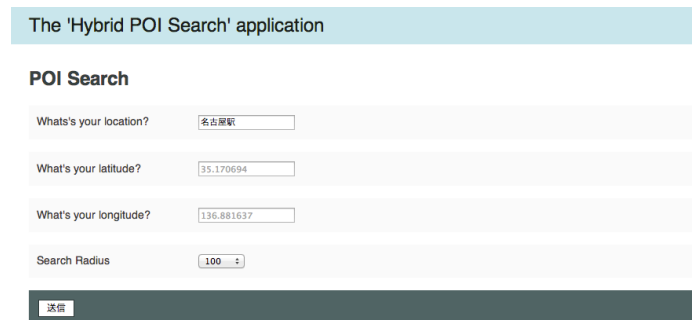


図 5 クライアントの現在位置入力画面

図 6 にクライアントが入力した位置情報を用いて、複数のサービスから取得した POI を表示した画面を示す。今回開発したプロトタイプシステムは現在、Hot Pepper, ぐ

るなび、食べログの3つの位置情報サービスから取得できた飲食店のPOIを表示する。地図表示にはGoogle Maps API[15]を用いており、取得できたPOIはマーカーで表示している。複数のサービスから取得できていることを可視化するために、マーカーの色分けを行なっている。ぐるなびは緑、食べログは青、Hot Pepprは赤に対応しており、1つのPOIに対して複数のサービスから取得できた場合は取得元のサービスの色を混合した色に対応している。図6の左部分はPOI名をリスト表示しており、クリックするとマーカーをクリックした時と同様の動作をする。図7にマーカーのクリック時に表示されるインフォメーションウィンドウを示す。これはPOIの名前・ジャンル・住所・電話番号情報・取得元サービスのアイコンを表示している。アイコンはクリック可能で各サービスへのリンクとなっている。図7では、POIの基本情報しか表示されていないが、クライアントが要求することで他の属性情報を表示可能である。インフォメーションウィンドウの下部にある編集ボタンから表示しているPOIの属性情報の編集画面に移ることができる。図8に編集画面を示す。今後は、POIの属性情報を編集画面から追加可能とし、クライアントのコンテキスト情報を元に対応した属性情報を表示可能な仕様に開発を進める予定である。

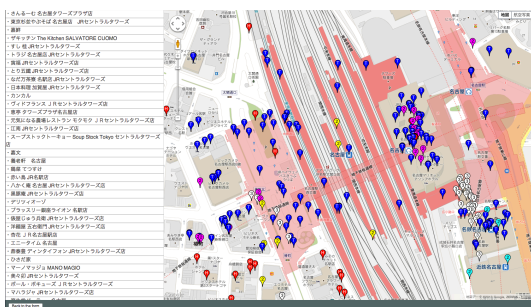


図6 POI表示画面

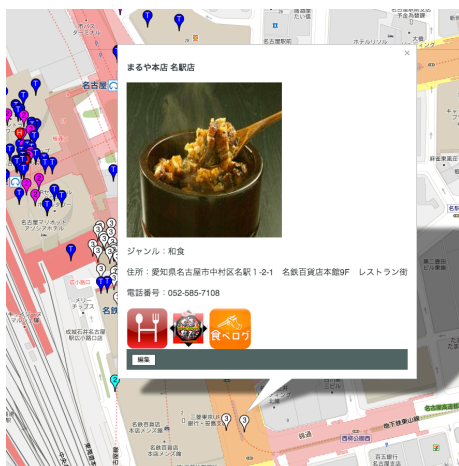


図7 POI詳細画面

EDIT_POI_DATA

HotPepper_ID :	<input type="text" value="xxxxxxxxxx"/>
ぐるなび_ID :	<input type="text" value="xxxxxxxxxx"/>
食べログ_ID :	<input type="text" value="xxxxxxxxxx"/>
名前 :	<input type="text" value="まるや本店"/>
緯度 :	<input type="text" value="35.16862977874623"/>
経度 :	<input type="text" value="136.885028033887"/>
住所 :	<input type="text" value="愛知県名古屋市中村区名"/>
建物名 :	<input type="text" value="名鉄百貨店本館"/>
フロア名 :	<input type="text" value="9F"/>
電話番号 :	<input type="text" value="052-585-xxxx"/>
カテゴリー :	<input type="text" value="和食"/>

図8 POIの属性情報編集画面

4. POI統合手法

複数の位置情報サービスから得られた多くのPOIを統合するために、POIの住所・電話番号・名前情報を用いてPOIの同定を行う。この3つの情報を用いる理由は、POIを活用したサービスの多くから取得できる情報であり、各POIに関して一意に決まる可能性が高いと考えられるからである。

まず、これらの情報のフォーマットを整えるため、属性情報の正規化を行う。住所情報に関しては、外部のサービスである住所正規化API[16]を活用する。例えば、住所情報(名古屋市中村区名駅3丁目14-16)をこのAPIにリクエストすると、(愛知県名古屋市中村区名駅3-14-16)の様に住所情報の中で足りない情報を補完し、フォーマットに即した丁番号表記に変換して住所を正規化する。住所の正規形には、建物名とフロア名は含まない。電話番号情報は国際番号表記であれば国内表記に変換し、また電話番号と関連性のない文字列が含まれていれば削除し、ハイフンを含めた形を正規形とする。名前情報に関しては正規化は行わない。

次に得られた複数のPOIを住所情報の文字列マッチングにより住所ごとにグルーピングをする。そして、各住所グループ内でPOIのすべての組み合わせにおいて同定をする。グループ分けをする理由は同定できるPOI同士は同じ住所情報を持つと考えられ、グループ内で同定を行うことで組み合わせの回数が減り、計算効率が向上すると考えられるからである。次に同定手法について述べる。

4.1 同定手法

同定をする際には、情報統合部内のPOI同定システムを用いる。2つのPOIを入力とし、同一のPOIであるかを出力する。まず、入力された2つのPOIの電話番号情報の文字列マッチングを行う。2つが同じPOIであれば、共通

の電話番号情報を保持しているはずである。電話番号情報が同じでない場合、入力している2つのPOIは異なっていると、この段階で判断することでPOI同定システムの処理速度を向上させる。同一であれば名前情報を用いて名前文字列判定を行う。また、一方のPOIに電話番号情報が含まれていれば、POI検索APIを活用したPOI検索判定を、また両方に含まれていなければ名前文字列判定を行う。上記の様にPOI同定システムは2つの判定手法を場合にに応じて使い分けており、最終的な判定は2つの判定手法により行っている。次に2つの判定手法について述べる。

POI検索判定：POI検索APIとは電話番号情報を用いてリクエストを送り、同じそれを持つPOIの情報を取得できるAPIとする。入力する2つのPOIをA・Bとし、Aの電話番号情報を用いてBの取得元のPOI検索APIからPOI:C(以下C)が得られたとする。取得元位置情報サービスが同じなので、CがBと同一のPOIの情報ならば、保持するPOIの属性情報やそのフォーマットは同一であると考えられる。POIに固有であると考えられる名前情報またはサービスが保持するID番号でBとCに対して文字列マッチングを行い、同一文字列であれば、AとBを同定する。この判定はAが電話番号情報を保持、かつBの取得元位置情報サービスがPOI検索APIに対応している必要がある。

名前文字列判定：2つのPOIのPOI名文字列から編集距離[17]と最長共通部分文字列長を求める。POI名文字列とは、POIの名前情報の文字列からPOI所在地文字列を除いた文字列を指す。名前文字列をスペースを区切りに複数の文字列に分割する。スペースで分割する理由としては、位置情報サービスから取得できたPOIの名前情報において、POI名文字列やPOI所在地情報がスペースにより区切られていることが多いことから、この区切られている部分文字列は何らかの意味を持っていると考えるからである。また、POI所在地文字列はその中で「店」を最後尾に含む部分文字列を指す。POI名文字列抽出の過程の例として、名前文字列「赤から 名駅西口店」の場合だと、POI名文字列として「赤から」が得られる。最後に、2つのPOIのPOI名文字列から求めた編集距離、最長共通部分列長とPOI所在地情報削除前の名前情報文字列の文字列長、編集距離、最長共通部分列長、真の判定結果の7つを特徴量として与え、機械学習により得られたアルゴリズムにより同定をする。機械学習を用いた理由としては、POIの名前情報にはPOI名と関連性のない文字列が含まれていたり、POI名の名前文字列は多様であり、判定時の閾値を一意に決めることが困難であるからである。

5. 評価と考察

提案する2つの同定手法を場合にに応じて使い分けるPOI同定判定システムが各同定手法を単独で使用するよりも効

果的であることを確認するために、評価実験を行った。機械学習ソフト「Weka」に正解：50件、不正解：450件のデータセットを与え、分類器には「J48」を用いた(編集距離：D、最長共通部分列長：L)。これより、Lが2以上かつDが8以下の場合とDが9以上ならばLが5以上である場合に同定するアルゴリズムを得た。このアルゴリズムを名前文字列判定の最終判定に用いる。

提案するPOI同定システムの評価を行った。評価データとして、名古屋駅を中心に半径1.0km以内に含まれる飲食店のPOIをHot Pepper、ぐるなび、食べログの3つの位置情報サービスから取得した。取得できたPOIは、Hot Pepper：569件、ぐるなび：484件、食べログ：1021件であり、統合可能POI数の真値は同定可能POIが555件、同定不可能POIが791件であることを目視により確認した。今回、POI検索判定・名前情報判定・2つの判定手法を複合して用いているPOI同定システムの3つの場合において評価実験を行い、評価の指標として精度と再現率を求めた。その結果を表3、4、5にまとめた。POI同定システムが適合率：1.0、再現率：0.91であり、最も良い結果であり、2つの手法を複合して用いた方が効果的であることが確認できた。

表3 POI検索判定

		真の結果	
		可	否
予測結果	可	413	38
	否	138	663

表4 名前文字列判定

		真の結果	
		可	否
予測結果	可	456	6
	否	95	695

表5 POI同定システム

		真の結果	
		可	否
予測結果	可	503	0
	否	48	701

表6 評価結果

	POI検索判定	名前文字列判定	POI同定システム
適合率	0.91	0.98	1.0
再現率	0.74	0.82	0.91

5.1 考察

名前文字列判定と POI 検索判定の 2 つの判定手法についての考察を述べる。

名前文字列判定の課題点を 3 点挙げる。1 点目は、最終的に同定している機械学習から得られたアルゴリズムでは、名前文字列長が 1 の場合は同定不可と判定されてしまうので、名前が 1 文字の店舗は判定ができない点である。2 点目は、名前情報文字列には店舗名と関連性の無い文字列が含まれているということである。特にぐるなびから取得した POI の名前情報に店舗のキャッチコピーや店舗のジャンルや設備情報が含まれていることが多かった。これらの文字列を含んでいない名前文字列と比較した時に、余分な文字列分が編集距離に反映され値が大きくなってしまい、同定不可能と判定される例が多数あった。3 点目は、店舗名の表示言語には対応できないという点である。具体例を表 7 に示す。Hot Pepper の名前情報はカタカナと英語が混合しているのに対して、ぐるなびは英語のみ、食べログはカタカナと漢字で表記されている。表記言語が違うことで編集距離が大きくなり、同定不可と判定される。

POI 検索判定の課題点を 2 点挙げる。1 点目は、POI 検索 API を使用する位置情報サービスによるが、1 日の Web API の使用回数は制限されている場合が多く、POI 検索判定を多用することは望ましくない。2 点目は 1 つの電話番号情報を複数の POI が持っている場合には対応できない点である。これは複数の POI の親会社と同じであったり、デパートのような商業施設が電話番号を管理している場合が該当する。同一の具体例を表 8 に示す。

表 7 店舗名の表記言語の違い

位置情報サービス名	名前情報
Hot Pepper	”ザキッチン The Kitchen SALVATORE CUOMO”
ぐるなび	”The Kitchen Salvatore Cuomo NAGOYA ～JR セントラルタワーズ 12 F～”
食べログ	”ザ キッチン サルヴァトーレ クオモ 名古屋”

表 8 電話番号が同じ店舗

位置情報サービス名	名前情報	電話番号
ぐるなび	名古屋 浅田	052-569-xxxx
食べログ	伊兵衛	052-569-xxxx

6. まとめと今後の課題

6.1 まとめ

本稿では、ユーザが利用したい POI の情報を個別の位置情報サービスからは網羅的には取得できない点、各位置情報サービスが持つ属性情報の違う点、記述フォーマットが統一されていない点、また誤った情報が含まれているとい

う 4 つの課題点を解決するために、複数の位置情報サービスから取得した POI の情報を統合的に利用可能とするプラットフォームを提案した。また、POI の情報統合するための同定手法を提案した。提案プラットフォームの実装例としてのプロトタイプシステムを Web アプリケーションとして実装した。また、提案する同定手法についての評価実験を行った。

6.2 今後の課題

POI の統合情報を RDF として保存するデータベースである RDF ストアを構築し、Linked Open Data として公開することを考えている。POI 数の網羅性と情報の正確性向上の為、対応サービスの追加と様々なオープンデータとの連携をする予定である。オープンデータの例としては、Open Street Map[18] が有力である。また、本プラットフォームを API として、公開・実運用することを考えている。

参考文献

- [1] Foursquare. <http://ja.foursquare.com/>
- [2] Google Maps. <http://maps.google.co.jp/>
- [3] Hot Pepper. <http://www.hotpepper.jp/>
- [4] 石田 和成. ジオメディアの分布調査および携帯端末を用いたジオローカルコンテンツ構築システムの開発. 情処研報. DD, [デジタル・ドキュメント] 2011-DD-80(11), 1-8, 2011
- [5] Google Places. <http://www.google.co.jp/landing/placepages/>
- [6] Yahoo!ローカルサーチ API. <http://search.olp.yahooapis.jp/OpenLocalPlatform/V1/localSearch>
- [7] ぐるなび. <http://www.gnavi.co.jp/>
- [8] 食べログ. <http://tabelog.com/>
- [9] Yahoo!地図. <http://map.yahoo.co.jp/>
- [10] 東田 圭介. クラウドソーシングを用いた POI 情報収集. 信学技報. AI, 人工知能と知識処理 111(447), 17-19, 2012-02-21
- [11] 鈴木 友基. クラウドソーシングによる屋内構造地図情報の構築と収集. 信学技報. MoMuC, モバイルマルチメディア通信 111(296), 1-6, 2011-11-10
- [12] Braun, Max. Context-aware collaborative creation of semantic points of interest as Linked Data. Master's thesis, University of Koblenz-Landau, Germany (2009).
- [13] KIM, Jong-Woo; KIM, Ju-Yeon; KIM, Chang-Soo. Semantic LBS: ontological approach for enhancing interoperability in location based services. In: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Springer Berlin Heidelberg, 2006. p. 792-801.
- [14] Google Geocoding API. <https://developers.google.com/maps/documentation/geocoding/?hl=ja>
- [15] Google Maps API. <https://developers.google.com/maps/?hl=ja>
- [16] 住所正規化 API. http://tou.ch/developer/api_all?uri=geo
- [17] 花田 博幸, 工藤 峰一. 編集距離による文字列の探索高速化に関する研究. 信学技報.PRMU, パターン認識・メディア理解 108(94), 41-45, 2008
- [18] OpenStreetMap. <http://www.openstreetmap.org/>