情報処理学会論文誌 Vol.60 No.2 549–560 (Feb. 2019)

事前検索を用いた多様性を考慮した高速なTwitter時空間検索

落合 桂一[1,2,a)] 鳥居 大祐[1] 深澤 佑介[1] 松尾 豊[2]

受付日 2018年5月7日, 採録日 2018年11月7日

概要：Twitter では今の話題を検索できる．近隣の話題を検索する場合，ユーザはスポット名などの適切な検索語を知らない場合もあり位置情報を検索クエリとして投稿を検索することが有用である．k 近傍検索では高速に検索できるが検索結果の多様性が低いという課題があり，地理的に近い投稿を範囲検索し，その結果を時間順にソートすることが有効である．しかし，検索する位置によって検索範囲を適切に設定しないと得られる投稿数が都市部と郊外で大きく異なり応答速度に影響がある．そこで，本稿では，多様性を考慮しながら検索ユーザの位置に基づいて検索範囲を設定することで周辺の投稿を高速に検索する手法を提案する．検索ユーザの位置に基づいて効率的に時空間検索するため，検索対象のスポットの位置で一定の投稿数を取得可能な最小半径を求めておく．ユーザが周辺の投稿を検索するときは，検索半径の初期値として事前検索で求めた半径を設定して検索する．約 100 万件のツイートを用いた評価を行い，提案手法が多様性を考慮したベースラインより 10.1%～29.0%高速に検索できることを確認した．また，主観評価実験により提案手法が k 近傍検索より次の行動の意志決定の参考になることを確認した．

キーワード：Twitter，時空間検索，位置情報，事前検索，多様性

Fast and Diversity-aware Spatio-temporal Search for Twitter using Pre-search

Keiichi Ochiai[1,2,a)] Daisuke Torii[1] Yusuke Fukazawa[1] Yutaka Matsuo[2]

Received: May 7, 2018, Accepted: November 7, 2018

Abstract: Microblogs such as Twitter can be used for searching trending topics. In the case of searching nearby tweets, it is useful to use the user's current location as a query instead of keywords, because the user may not know appropriate spot name or event name. K-Nearest Neighbor query can search quickly, however, there is diversity problem. It is effective to search geographically close posts by range query and sort the results in chronological order. However, if fixed range query is used for every location, it has a problem with response time. Therefore, we propose a fast and diversity-aware method for searching nearby tweets using location adaptive range query. To grasp nearby topics, it is needed to obtain a certain number of tweets as search result. To efficiently search tweets based on the user's location, we need to know a minimum radius which contains a certain number of tweets. Nearby tweets are searched based on spot-associated radius which is explored in advance as the initial range. Our evaluation on one million tweets shows that the proposed method is about 10.1%-29.0% faster than the diversity aware baselines. In addition, subjective evaluation shows that the proposed method is more valuable than kNN query for activity planning.

Keywords: Twitter, spatio-temporal search, location, pre-search, diversity

[1] 株式会社 NTT ドコモ
NTT DOCOMO, INC., Yokosuka, Kanagawa 239–8536, Japan
[2] 東京大学
The University of Tokyo, Bunkyo, Tokyo 113–8656, Japan
[a)] ochiaike@nttdoomo.com

1. はじめに

Twitter や Facebook などの SNS（Social Networking Service）が普及している．SNS ではリアルタイムに実世界についての投稿が行われるため，注目されているイベントや

スポット，出来事に関する投稿から今の話題を知ることができる．総務省の調査[*1]では，20 代のスマートフォンユーザでは，今何が流行っているかを知るために SNS を検索するといった利用が行われていることが報告されている．たとえば，Twitter ではイベント [1]，商業施設や観光スポットなどの施設（Point-of-Interest，以下 POI）[2]，電車の運行状況 [3] などに関して現在の状況や話題を投稿から知ることができる [4]．

ここで，本研究で想定する SNS およびユーザについて説明する．本研究では Twitter を対象とし，ユーザが外出先で次の行動や目的地を決めるために，ユーザの現在地周辺の商業施設や観光スポットなど POI に言及している投稿（以下，POI 関連ツイート）から即時性の高い情報を得るために投稿を検索するという状況を想定する．有用な検索システムを提供することでユーザの行動支援を行う．ユーザは必ずしもスポット名などの適切な検索語を知っているとは限らないため近隣で注目されているものを調べる方法の 1 つとして，スポット名などのキーワードの代わりに位置情報（緯度，経度）をキーに投稿を検索することが有用である．このような状況では，検索システムに以下の要件が求められる．

(1) **投稿日時が新しい投稿が好適**
ユーザが Twitter を情報源とする利点は即時性であるため，投稿日時が新しい投稿がユーザにとって有用であると考えられる．そのため，検索結果を時間順でソートすることが有用であると考えられる．

(2) **ユーザにより近い POI に言及している投稿が好適**
ユーザの現在地に近い方が訪問しやすく，遠い POI が検索されても訪問する可能性は低く，ユーザが訪問可能な許容範囲があると考えられる [5]．

(3) **検索結果に多様性がある**
周辺の話題を知るためには単一の POI に関する投稿より，複数の POI に関する投稿を閲覧できる方がより周辺の話題を把握しやすい．

(4) **高速に検索できる**
一般に検索システムでは応答時間が短い方がユーザにとって利便性が高い．

位置情報付きの投稿のような時空間情報データベースの検索方法に，k 近傍検索（k-Nearest Neighbor query）と範囲検索（range query）がある [6], [7]．要件 (1)，(2) を満たす方法として，POI 関連ツイートを距離でソートし上位 N 件を取得し，それを時間順にソートする方法（k 近傍検索）が考えられる．k 近傍検索は文献 [8] をはじめ多くの研究が行われており高速に検索することができる．しかし，この方法ではユーザの現在地から最も近い POI（最近傍 POI）に検索結果として提示する数（検索結果件数）以上の投稿があるときは，最近傍 POI の投稿のみが検索結果となり，最近傍 POI 以外の近隣の POI に関する投稿が新しい投稿でも検索結果に含まれないという課題がある．これは POI 関連ツイートには POI の緯度経度が付与されており，同一緯度経度の投稿が多いことに起因する．たとえば，都市部の有名な観光スポットでは投稿が多く，検索結果がその観光スポットに関する投稿のみになる．一方，多様性の要件 (3) を満たすため，一定範囲内の POI を検索し，各 POI に言及している投稿を 1 つずつ取得するという方法が考えられる．この方法では，頻繁に言及される POI では最新の投稿が得られるが，あまり投稿がない POI に関しては古い投稿しか得られず，検索結果全体としては最新の投稿が少なくなる．そのため，要件 (1)〜(3) を満たすため地理的に近い投稿を範囲検索により取得し，その結果を時間順にソートすることが有効であると考えられる．このとき，地理的な検索を行う範囲の決定方法が検索の性能に影響する．検索半径を全国一律の固定値にした場合，固定値の半径が短いと都市部では様々な POI が密集しているため多様な検索結果が得られるが，郊外でも同じ半径を設定すると検索結果がない場所が多くなったり多様性が低い検索結果になりやすい．一方，半径を長く設定すると場所によらず多様性が高い検索結果が得やすいという長所があるが，ユーザが検索した位置から投稿位置までの距離が長い投稿が含まれたり都市部では多くの投稿が検索範囲内に含まれ応答時間が長くなるという短所がある．

そこで本研究では，検索ユーザの位置に基づいて検索範囲を設定することで多様性を考慮しながら周辺の投稿を高速に検索する手法を提案する．地理的に近い投稿を範囲検索により取得することで要件 (1)〜(3) を満たす．要件 (4) を満たすために，事前検索により各位置で検索結果件数の投稿が得られる半径を探索しておくことで冗長な検索範囲を設定することを防ぎ検索を高速化する．すべての場所で事前検索を行うことは現実的ではないため，効率的に事前検索を行う必要がある．本研究では POI 関連ツイートを検索対象とするため POI の位置で事前検索する．実際にユーザが投稿を検索するときは，事前検索で求めておいた検索結果件数の投稿が得られる最小の半径（以下，最小半径）を周辺の投稿を検索する範囲検索の初期値として検索する．すなわち各場所で検索半径のインデックスを作成しておくことで多様性を考慮しながら高速に検索を行う．

本研究の貢献は以下のとおりである．

- POI の位置で事前に検索した結果を利用し，位置に応じて検索半径を設定することで多様性を考慮しながら高速に投稿を範囲検索する手法を提案した．
- Twitter を対象として k 近傍検索より範囲検索の方が多様性が高いことを検証した．そのうえで，提案手法が多様性を考慮した手法において高速に検索できることを確認した．提案手法ではベースライン手法より約

[*1] http://www.soumu.go.jp/johotsusintokei/linkdata/h29_01_houkoku.pdf

10.1%～29.0%高速に検索できる．

- 主観評価実験により，k 近傍検索より多様性のある検索となる範囲検索を行うことで外出の参考になる投稿が得られることを明らかにした．

本稿の構成は以下のとおりである．初めに 2 章で関連研究を述べ，次に 3 章で提案手法を説明する．4 章で提案手法の有効性を確認するために行った評価実験について述べ，5 章で本研究のまとめと今後の課題を述べる．

2. 関連研究

2.1 時空間データベース

本研究では位置と時刻をキーに投稿の検索を行うため，時空間データベースを対象にした近傍検索の従来研究との差異について述べる．時空間データを対象にした近傍検索の研究では，以下の研究が主に行われている．

(1) k 近傍検索（k-Nearest Neighbor query; kNN query）および範囲検索（range query）の研究とその拡張
(2) 検索対象のデータが時間経過と共に発生することを対象とした研究

2.1.1 k 近傍検索・範囲検索とその拡張

空間情報や時間情報のデータベースへの問合せの種類として代表的な方法に k 近傍検索と範囲検索がある [6], [7]．検索キーが空間情報のみの場合は，k 近傍検索で検索結果の件数を指定して距離が近いデータを検索できる．空間情報を MBR（Minimum Bounding Rectangle）を用いて表現し R-tree の木構造でデータを保持することで空間情報を対象として k 近傍検索を高速に行うことができる．k 近傍検索に関する研究は Roussopoulos らの研究 [8] をはじめ多くの研究がある．しかしながら，本研究では空間情報だけでなく時間情報も考慮する必要があるため時空間での近傍検索となる．時空間での近傍検索は，地理的な近さと時間的な近さを直接比較できないことや，地理的，時間的な近さに重み付けしてクエリとの関連性を計算し Top-k を検索する場合でも，全データに対して関連性を計算する必要があり計算コストが非常に高くなる [4], [9]．また，1 章のシステム要件で述べたとおり，地理的な近さで k 近傍検索を行い上位 N 件の投稿を取得し，それを時間順にソートして検索結果の件数を取得する方法もあるが，多様性が少ないという課題がある．

2.1.2 検索対象のデータが時間経過と共に発生することを対象とした研究

Magdy らの研究 [10] では半径 R，時間 T 内の投稿に対して，時空間の近さで各投稿に重み付けしたスコアを計算し Top-k の投稿を検索する手法を提案している．この手法では半径 R と時間 T は固定値であり，本研究では半径 R を検索する場所により適応的に変化させる点が異なっている．Nepomnyachiy ら [4] は Twitter などの時空間情報を検索するための 3W（What，Where，When）という時空間検索フレームワークを提案した．この研究では，検索時に空間と時間の範囲を指定して検索し，テキストの転置インデックスと組み合わせることで時空間でのテキスト検索を行っている．空間的なインデックスを構築するときに Magdy らの研究 [10] と同様に各矩形の最大投稿数を決め，矩形のツリー構造ができるだけ浅くなるようにインデックスを作成する．検索時は地理的な検索範囲がユーザによって決められるため，検索範囲を自動的に決定することは行っていない．

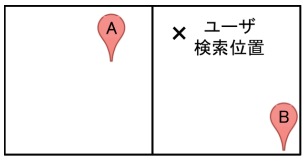

図 1 グリッド単位での計算で問題となる場合
Fig. 1 Example of ineffective case about grid-based method.

このように SNS のように時間経過にともないデータが発生する状況での時空間検索では，時間および空間の範囲検索を行い，その範囲内の投稿を検索クエリとの関連性を評価値として Top-k 検索している．場所によらず一定の範囲で範囲検索を行うと 1 章で述べたとおり検索性能に影響がある．一方，本研究では，空間情報は一定数の投稿が得られる範囲での範囲検索を行い，時間情報に対しては k 近傍検索を行うことで時空間検索を行っている．

2.2 Shaw らの研究

本研究に関連する研究として Shaw らの研究がある [11]．Shaw らは Foursquare[*2]のサービスにおいてユーザがチェックインする POI を高速に検索する手法を提案した．この手法では，グリッド単位に POI 密度を事前に計算しておき，検索結果として得られる POI 数が一定数以上になる範囲（半径）を POI 密度から計算する．Shaw らの研究では 150 カ所以上の POI が検索結果として得られる半径を，約 250 m 四方のグリッドごとの POI 密度から計算した．Shaw らの問題設定は，一定数の地理情報を高速に検索するために位置によって検索範囲を変えるという内容であり，本研究と類似している．この手法をツイート検索に応用し，メッシュごとに POI 関連ツイートの密度を計算し一定数の投稿が得られる半径を推定することで POI 関連ツイートを高速に検索できると考えられる．しかしながら，グリッド単位で密度を計算する方法では，グリッドの端にデータがある場合に問題がある．**図 1** の例で説明する．図中の A，B に POI があり，×の位置でユーザが検索した場合を想定する．ここで，A に言及している投稿数が多く，B に言及している投稿数が少ないとする．このような状況では，ユーザ検索位置のグリッドの投稿密度は低くなり広い半径を設定し投稿を検索する．しかし，ユーザ検

*2 https://www.swarmapp.com/

索位置から距離が近い A に言及している投稿数が多いため応答時間がかかる．そのため，グリッド単位ではなく，検索位置から近い POI の言及数に基づいて検索範囲を決めることが有効であると考えられる．

2.3 検索システムの高速化

Web 検索では，検索の応答時間を短縮するためにクエリと検索結果をキャッシュしておく方法がある [12]．キーワード検索では同一クエリとなることが多いが，位置情報（緯度，経度）をクエリとした場合，まったく同一の位置で検索することは少ないためキャッシュによる高速化は適さないと考えられる．

3. 提案手法

提案手法の処理の流れを**図 2** に示す．提案するシステムはジオタギング，事前検索，クエリ処理の 3 つのモジュールで構成される．ジオタギングモジュールでは，各投稿に対して POI との関連付けを行う．POI 名称と投稿で単純な文字列マッチを行うと地名に曖昧性がある場合，誤った POI に関連付けてしまうため文献 [2] の手法を用いて地名の曖昧性解消を行い，Twitter データベース（以下，Twitter DB）にツイートを保存する．ジオタギングモジュールではツイートと POI を関連付けるため，検索対象となる投稿はすべて POI に関連する投稿となり緯度経度が付与される．これにより，スポット名称によるキーワードでの検索をユーザの位置（緯度，経度）での検索に置き換えることができる．事前検索モジュールでは，検索結果件数 Tw_{min} を取得できる最小半径を探索する．事前検索では仮想的にユーザが POI の位置にいるとして，その位置で検索結果件数を取得できる最小半径を調べ POI の位置と対応付けて事前検索半径データベース（以下，事前検索 DB）に最小半径を保存する．ユーザが周辺の投稿を検索するときは，クエリ処理モジュールがユーザの位置を受け取り，事前検索 DB に対して投稿を検索する半径の初期値を問い合わせる．このときユーザの位置に最も近い POI に対応する事前検索半径を取得する．そして，クエリ処理モジュールは Twitter DB に対してユーザの位置と半径の初期値として事前検索半径を送り，ユーザの周辺の投稿を検索する．本章では，ジオタギング，事前検索，クエリ処理の各処理について詳細を説明する．

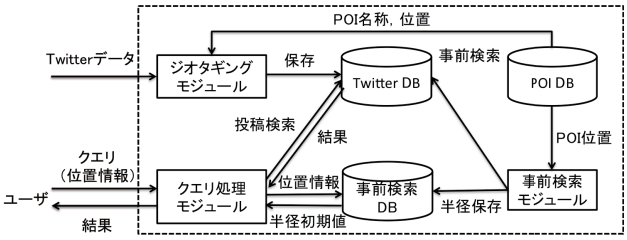

図 2 提案手法の流れ
Fig. 2 Overview of the proposed method.

3.1 ジオタギング

位置に関連するツイートは大きく 2 つある．1 つはユーザが投稿した位置の緯度経度（ジオタグ）を付加して投稿されたツイート（ジオタグ付きツイート），もう 1 つはツイート本文をテキスト解析し，地名やスポット名など POI 名称を抽出して位置と関連付けたツイート（POI 関連ツイート）である．本研究では検索対象として POI 関連ツイートのみを対象とする．理由は以下の 2 点である．

(1) ジオタグ付きツイートより，ツイート本文に地名が含まれる POI 関連ツイートの割合の方が多く [13], [14]，周辺情報の把握という観点では量が多い POI 関連ツイートが適している．
(2) ジオタグ付きツイートは必ずしも投稿した位置に関する話題だけが投稿されるわけではなく日常会話なども含まれノイズとなる可能性があるため [15]，周辺の話題を知るという目的に対して情報の質の観点で POI 関連ツイートがより適している．

ツイート本文を解析し位置と関連付けるため，ツイート本文と POI DB の POI 名称とのマッチングを行う．あらかじめ地名やスポット名など POI 名称と緯度経度の対応関係を POI DB として持ち，ツイートを形態素解析し，形態素と POI DB のマッチングにより位置と関連付ける．たとえば，「東京タワー」という文字列がツイートを形態素解析して得られた場合，POI DB に保持している東京タワーの緯度経度を該当ツイートに付与する．テキスト解析で位置と関連付ける場合は地名の曖昧性解消が課題となる場合がある．

地名の曖昧性は 2 種類存在する [16]．1 つは Geo/Non-geo 曖昧性と呼ばれ，地名と同じ表記で地名以外の意味を持つ単語の曖昧性である．たとえば「品川」という表記は，地名としても人名としても使われる．もう 1 つは Geo/Geo 曖昧性と呼ばれ，同じ表記の地名が複数の地理的な場所に存在する単語の曖昧性である．たとえば「日本橋」という表記の地名は東京と大阪に存在する．

Geo/Non-geo 曖昧性の解消は CRF（Conditional Random Field）を用いた固有表現抽出を利用したり [16], [17]，各場所の特徴語との共起を利用する手法 [2] がある．一方，Geo/Geo 曖昧性の解消には「同一文書に現れる地名は地理的に近い場所を示す」という仮定のもと，地理的に近い地名（近隣地名）や，地名辞書の階層が近い地名（たとえば市区町村名であれば 1 つ上の階層の都道府県名や同一都道府県内の市区町村名など）との共起や特徴語との共起により曖昧性解消する方法が多い [2], [14], [16], [17]．

本研究では Geo/Non-geo 曖昧性の解消には特徴語の共起を用いる手法 [2] を利用し，Geo/Geo 曖昧性には近隣地名と特徴語の共起を用いた [2], [16]．文献 [2] の手法では，再現率 0.53，適合率 0.93 の精度で地名の曖昧性を解消できる．

Algorithm 1 Pre-Search Procedure.

Input: POI Location $l = (lat, lng)$
Output: Radius R
1: $TwList \leftarrow \emptyset$
2: Set search radius $R \leftarrow R_{init}^{pre}$
3: $TwList \leftarrow$ search tweet at (lat, lng) and radius R
4: **while** $|TwList| \leq Tw_{min}$ and $R < R_{max}$ **do**
5: $R \leftarrow R + R_{extend}$
6: $TwList \leftarrow$ search tweet at (lat, lng) and radius R
7: **end while**
8: **return** R

Algorithm 2 Nearby Tweet Search Procedure.

Input: User Location $l = (lat, lng)$
Output: Tweet List $TwList$
1: $TwList \leftarrow \emptyset$
2: Set search radius $R \leftarrow R_{init}$
3: $TwList \leftarrow$ search tweet at (lat, lng) and radius R
4: **while** $|TwList| \leq Tw_{min}$ and $R < R_{max}$ **do**
5: $R \leftarrow R \times 2$
6: $TwList \leftarrow$ search tweet at (lat, lng) and radius R
7: **end while**
8: **return** $TwList$

3.2 事前検索

事前検索では，Tw_{min} 以上の投稿数を得られる最小半径を探索する．伊藤ら [18] は，都市に配置したセンサネットワークを対象に，センサ値を補間するため可変半径法 [19] を応用し一定数のセンサを取得する方法を提案している．可変半径法とは，ある地点を中心として，与えられた距離を半径の初期値としてデータ点を検索し，この範囲に必要数のデータ点が存在しない場合は半径を拡大し検索するというデータ点の抽出手法である．伊藤らの手法では，可変半径法を応用し，与えられた半径の初期値で周辺のセンサを検索し，補間に必要な一定数に達しなければ初期半径を 2 倍，3 倍と増やし検索範囲を拡大していく．本研究でも可変半径法を応用して一定数の投稿を取得可能な半径を探索する．Algorithm 1 に事前検索の処理を示す．この処理を POI データベースに存在するすべての POI に対して実行する．本研究では POI 関連ツイートを検索対象とするため，各 POI の位置（緯度，経度）で事前検索する．事前検索は，たとえば 1 日に 1 回など定期的に実行される．ここで R は周辺投稿を検索する半径，$TwList$ は緯度 lat，経度 lng の位置で半径 R で投稿を検索したときに検索された結果の投稿のリスト，R_{init}^{pre} は事前検索の半径初期値，R_{extend} は投稿が検索結果件数に達しなかった場合に拡大する半径の増分，R_{max} は周辺投稿を検索するときの最大半径を示す．提案するシステムでは，1 章で述べたとおり，ユーザが許容できる最大距離があると考えられるため R_{max} を設定している [5]．

3.3 クエリ処理

Algorithm 2 に事前検索半径を用いてユーザの周辺の投稿を検索する処理の流れを示す．最初にユーザの現在地に最も近い POI を検索し，事前検索 DB から最近傍 POI に対応する半径の初期値（最小半径）R_{init} を取得する．そして，周辺の投稿をユーザの位置を中心として半径 R_{init} で検索する．R_{init} で周辺投稿を検索し投稿数が検索結果件数 Tw_{min} に達しない場合，検索する半径を 2 倍に拡大し再検索を行う．この処理を検索結果件数 Tw_{min} が得られる，もしくは最大半径 R_{max} に達するまで検索範囲を拡大する．ここで半径を 2 倍にしているのは，参考とした可変半径法を応用した先行研究 [18] で半径拡大の初期値が 2 倍だったためである．また，先行研究 [18] では 2 倍，3 倍と倍数を 1 ずつ増加させていくが，本研究では 2 倍ずつ（2 倍，4 倍，8 倍 $\cdots$）拡大するアルゴリズムを採用する．提案手法のクエリ処理では，ユーザに高速に検索結果を提供する必要がある．そのため，一定数の投稿が得られるできるだけ狭い範囲となる半径を正確に探索するより，一定数以上の投稿を取得できる半径に速く到達することを重視し指数関数的に半径を拡大するアルゴリズムとした．

4. 評価実験

本章では提案手法の性能を評価するために行った実データを用いた実験について説明する．評価は 1）定量評価，2）定性評価，3）事前検索の実用性評価，4）ユーザビリティ評価の 4 つを行う．定量評価では，単純な手法である場所によらず半径初期値を固定した手法（B1），Shaw らの手法 [11] により POI 関連ツイートの密度に基づいて半径初期値を決める方法（B2），k 近傍検索（B3）をベースラインとして比較する．定性評価では，検索位置によって半径設定が変化していること，および検索結果が周辺の話題を把握するという検索意図に適しているかを確認する．事前検索の実用性評価では，事前検索の計算コストを測定し，実用的な計算コストで実現可能か検証する．ユーザビリティ評価では本研究で想定する利用シーンでの提案手法の有用性を被験者実験により評価する．

4.1 定量評価指標

本研究の評価指標として，1）多様性，2）平均応答時間，3）投稿検索時の半径設定誤差の 3 つを用いる．1 章で述べたように k 近傍検索を用いる手法は高速に検索できるが検索結果の多様性に課題があった．そのため，範囲検索を用いた提案手法で多様性の課題が解決されていることを確認するための評価指標とする．具体的には，検索結果に含まれる平均 POI 数を指標とする．これは 1 章で述べた要件 (3) に対応する．また，要件 (4) に対応した評価指標として検索の応答時間を用いる．本研究ではユーザの現在地に近い投稿を検索することも目的としているため，遠い投稿を高速に検索できても有効ではない．そのため，検索した

ときの半径（検索対象の投稿までの距離）も評価指標として利用する．半径設定の正解データとして評価に用いたジオタグ付きツイートの各位置で半径初期値を 100 m として 100 m ずつ半径を拡大し，検索結果件数を得られた半径を正解データとし，正解データと各手法で設定した半径の平均絶対誤差を範囲設定の評価指標とした．なお，本研究では 1 章で述べた 4 つの要件を満たすシステムの提案を目的としており，システム要件に投稿時間の新しさがあるが，各手法において時間順にソートしているだけであるため投稿日時の新しさは評価指標とはしていない．

4.2 実験条件

4.2.1 パラメータ設定

実験において各パラメータは以下のとおり設定した．事前検索の半径初期値は $R_{init}^{pre} = 500\,\mathrm{m}$，事前検索における半径増分は $R_{extend} = 100\,\mathrm{m}$，ユーザ検索時の半径初期値は提案手法，ベースラインともに $R_{init} = 500\,\mathrm{m}$ とした．また，事前検索およびユーザ検索時の最大半径は $R_{max} = 10\,\mathrm{km}$，50 km の 2 パターン，検索結果件数は $Tw_{min} = 10$，30 の 2 パターンの合計 4 パターンで実験を行った．検索結果件数を 10，30 に設定したのは，一般的な商用の Web 検索エンジン（Google や Bing）での検索では，デフォルトの検索結果件数が 10 件となっていること，また，Twitter 検索の既存研究 [20] における評価で P@10，P@30 を指標としていたためである．最大半径の値は，文献 [5] において都市部と郊外でユーザが施設を訪問するときの許容可能な最大距離に関するアンケートを行った結果，都市部では平均 11.3 マイル，郊外では 24.65 マイルであった．文献 [5] は日本国外でのアンケートであるため，値そのものを日本に適用できるかは不明であるが，一般に都市部では最寄りの施設までの距離が短く郊外では長いという傾向は国によらず同じ傾向と考えられる．そのため，都市部に適した範囲が狭い設定と，郊外に適した範囲が広い設定の 2 パターンで検証を行うため 10 km，50 km とした．

4.2.2 データ

評価には 2016 年 4 月 2 日から 4 月 3 日の間に投稿された日本語の Twitter データを収集し，POI との関連付けを行った 925,418 件の POI 関連ツイートを用いた．これらのツイートには Foursquare などのチェックインアプリケーションを通したツイート，宣伝や広告のツイートも含まれる．チェックインアプリケーションからの投稿でもユーザがコメントを入力したものは周辺情報を知るうえでは有益であると考えられる．また，定型文となるもの（具体的には「I'm at ●●」など」）を集計したところ 10,569 件（約 1.1%）であり，チェックインしただけの投稿の有無が検索結果に与える影響は少ないと考えられるため，チェックインアプリケーションを通じた投稿もすべて検索対象に含めている．また，宣伝や広告のツイートの中に，これからイベントを開催するというものや，タイムセールを行うなどのツイートを検索しているユーザにとって有益となりうる内容も含まれるため検索対象に含めている．

4 月 3 日にユーザが位置情報を基に投稿検索したという状況を想定し評価を行った．そのため，前日の 4 月 2 日の投稿を対象に POI の位置で事前検索を行った．評価対象のエリアとして，東京都，神奈川県，大阪府の 3 都府県において，東京 23 区および政令指定都市を都市部，それ以外の市町村を郊外と設定した．ユーザの位置情報には，ランダムな緯度経度を生成して利用することもできるが，人がよく立ち寄る場所の偏りを反映するため，2016 年 3 月 28 日から 4 月 3 日の間に投稿されたジオタグ付きツイートから緯度経度が評価対象エリア内のツイートの位置を利用した．各エリア 1,000 件のジオタグ付きツイートをランダムサンプリングし，ジオタグ付きツイートの位置でユーザが検索したと仮定し評価を行った．各都府県で都市部と郊外の 2 つのエリアがあるため，合計 6,000 件のジオタグ付きツイートをユーザの位置として利用した．事前検索の場所を決めるための POI DB として，NTT ドコモが提供する「ご当地ガイド」という観光向けアプリで利用されている日本全国の観光スポットおよび駅データ.jp [*3]の駅名を合わせ約 4 万件のスポットを利用した．

4.2.3 Shaw らの手法（B2）の実装詳細

ここで，Shaw らの手法における半径の計算方法を説明する．文献 [11] では半径を計算する手順が以下のように述べられている．

(1) ユーザ位置におけるグリッドの ID を計算．
(2) ユーザ位置のグリッド ID をキーに POI 密度を参照し，一定数（文献 [11] では 150 個）の POI が検索できる半径 r を推定．
(3) 半径 r で POI を検索．

ここでグリッドの一辺を l とすると POI 密度 d は次式となる．

$$d = \frac{\#\text{ of POIs within grid}}{l^2} \tag{1}$$

期待する検索結果の POI の個数を n とすると

$$n = \pi r^2 \times d \tag{2}$$

よって，式 (1)，(2) より半径 r は次式となる．

$$r = l \times \sqrt{\frac{n}{\#\text{ of POIs within grid} \times \pi}} \tag{3}$$

Shaw らの研究では POI 検索を行っていたため POI 数となっている部分を本研究では投稿数に置き換える．また，本研究ではグリッドとして地域メッシュ[*4]を用いた．Shaw らの研究では Google がオープンソースで提供する S2 Geometry Library [*5]で定義されている S2 cell と呼ばれる階

*3 http://www.ekidata.jp
*4 http://www.stat.go.jp/data/mesh/
*5 https://github.com/micolous/s2-geometry-library

```
SELECT * FROM
  (SELECT poi_id, tw_id, time, tw_text FROM TW_TABLE
  ORBDER BY tw_geom <-> query_loc_geom LIMIT N) A
WHERE ST_DWithin(tw_geom, query_loc_geom, max_dist)
ORDER BY time desc LIMIT Tw_min
```

図 3 k 近傍検索の疑似コード
Fig. 3 Pseudo-code of kNN query.

層的な構造を持つ平行四辺形の領域をグリッドとして利用している．投稿密度を計算するためのメッシュサイズは，文献 [11] に従い 250 m 四方とした．

4.2.4 k 近傍検索（B3）の実装詳細

k 近傍検索では空間情報に対して k 近傍検索を行い近い順に N 件の投稿を取得し，その結果を時間順にソートする．**図 3** に k 近傍検索の疑似コードを示す．本研究では実装に PostGIS を利用した．疑似コードの `tw_geom` は検索対象の投稿位置（緯度, 経度）を PostGIS で位置情報を扱う geometry 型に変換したデータ， `query_loc_geom` は検索ユーザの位置を geometry 型に変換したデータ， `max_dist` は R_{max}， `Tw_min` は Tw_{min} を示す．サブクエリの時点で投稿数を Tw_{min} に絞るとユーザの最近傍 POI と距離があまり変わらない POI に関する投稿で時間が新しいものでも検索されない可能性があるためサブクエリでは Tw_{min} より多くの投稿を抽出し時間順にソートするところで Tw_{min} 件に絞る．実験では $N = a \times Tw_{min}$ とし，a の値を変えて実験を行った．a の値が小さい場合は距離が近いもののみが対象となるため多様性が低くなる．一方，a の値が大きいと距離が離れているものも含めて時間順にソートされるため多様性が高くなる．多様性は検索対象となる距離に影響されるため，地理的な範囲検索を行う提案手法や B1，B2 と検索範囲を揃えるため a のパラメータを調整した．$a = 1, 2, 5$ として検証を行い，検索結果として得られる POI 関連ツイートと評価用のジオタグ付きツイートの位置（仮想的なユーザ位置）の距離の平均値を計算した．その結果，都市部では $a = 5$，郊外では $a = 2$ の場合が提案手法や B1，B2 と同程度であったため評価対象エリアごとに a を変えて以降の実験を行った．

4.2.5 実験環境

評価に利用したマシンは Windows7 Professional，Intel(R) Xeon(R) CPU @ 2.60 GHz，RAM 64 GB である．データベースとして PostgreSQL 9.4.1，PostGIS 2.1 を利用し，空間インデックスを有効にして実験を行った．Shaw らの研究 [11] ではミドルウェアとして Solr [*6] を用いていたが，本研究ではアルゴリズムの差を検証するため Shaw らの手法も PostgreSQL および PostGIS で実装した．

[*6] http://lucene.apache.org/solr/

表 1 各パラメータでの検索結果に含まれる平均 POI 数（都市部）
Table 1 Mean POI number for each parameter (urban area).

手法	パラメータ (Tw_{min}, R_{max})			
	(10, 10)	(10, 50)	(30, 10)	(30, 50)
提案手法	1.855†	1.855†	2.780†‡	2.780†‡
固定値（B1）	1.864	1.864	2.731	2.731
Shaw らの手法 [11]（B2）	1.864	1.864	2.743	2.743
k 近傍検索（B3）	1.522	1.522	2.478	2.477

表 2 各パラメータでの検索結果に含まれる平均 POI 数（郊外）
Table 2 Mean POI number for each parameter (suburb area).

手法	パラメータ (Tw_{min}, R_{max})			
	(10, 10)	(10, 50)	(30, 10)	(30, 50)
提案手法	1.747†	1.749†	2.979†§*	2.991†♯
固定値（B1）	1.739	1.743	2.904	2.919
Shaw らの手法 [11]（B2）	1.739	1.743	2.908	2.924
k 近傍検索（B3）	1.565	1.574	2.605	2.621

4.3 定量評価

はじめに多様性に関する定量評価の結果について述べる．**表 1**，**表 2** に都市部および郊外における検索結果に含まれる平均 POI 数の結果を示す．表 1 において † は提案手法と B3 で有意水準 0.1%で有意差があること，‡ は提案手法と B1 で有意水準 5%で有意差があることを示す．また，表 2 においては，† は提案手法と B3 で有意水準 0.1%で有意差があること，§ は提案手法と B1 で有意水準 1%で有意差があること，∗ は提案手法と B2 で有意水準 1%で有意差があること，♯ は提案手法と B2 で有意水準 5%で有意差があることをそれぞれ示す．1 章で述べたとおり都市部および郊外のいずれのパラメータでも k 近傍検索（B3）では多様性が低いという結果であった．また，表 1 より都市部においては，最大半径 R_{max} が 10 km の場合と 50 km の場合で平均 POI 数が同じになった．このことから，都市部では 10 km 以内で一定の検索結果数 Tw_{min} が取得できていると考えられる．一方，表 2 より郊外では最大半径 R_{max} を変えることで平均 POI 数が変化しており，一定の検索結果数 Tw_{min} を取得するためには 10 km 以上の範囲を検索する必要がある場合があると考えられる．都市部および郊外のどちらの場合も，提案手法は B1，B2 と Tw_{min}=10 の条件では有意な差があるとはいえない結果となった．これは 1 章で述べたように地理的な範囲検索を行った後に時間順にソートするという処理の流れが提案手法，B1，B2 では共通しており同じような特性になったと考えられる．

次に応答時間の評価結果について述べる．**表 3** に検索結果件数 Tw_{min} および最大半径 R_{max} の各パラメータを変化させた全組合せ（4 パターン）全体の応答時間の平均，分散，最大値の結果を示す．前述の多様性の評価結果および後述する 4.6 節のユーザビリティ評価から k 近傍検索ではなく範囲検索することが多様性を考慮した今回のシステ

表 3 各手法の応答時間の平均，分散，最大（ms）
Table 3 The results of mean, std and maximum response time of each method.

手法	都市部			郊外		
	平均	分散	最大	平均	分散	最大
提案手法	**85.4**	66.6	**800**	**52.3**	11.7	**237**
固定値（B1）	95.0	67.6	830	61.6	11.7	252
Shaw らの手法 [11]（B2）	110.5	79.5	2058	73.7	13.1	509

表 4 各手法の半径の平均絶対誤差（m）
Table 4 The result of mean absolute error of each method.

手法	都市部	郊外
提案手法	105.9	427.2
固定値（B1）	101.5	435.9
Shaw らの手法 [11]（B2）	106.4	435.9

ム要件に適していると考えられるため，応答時間の評価では k 近傍検索（B3）以外のベースライン手法と比較を行った．太字はもっとも性能がよかった結果を示す．都市部および郊外のどちらの場合も多様性が同程度であった B1，B2 を対象に提案手法との差を t 検定したところ有意水準 1%で有意な差があった．B1，B2 に対しては提案手法の方が 10.1%〜29.0%高速に検索できる．また，分散，最長応答時間も比較手法と比べて，都市部，郊外ともに最も小さくなっており，提案手法は有効である．

次に，**表 4** に検索結果件数 Tw_{min} および最大半径 R_{max} の半径の平均絶対誤差の実験結果を示す．郊外エリアにおいては最近傍 POI を利用する提案手法が最も狭い半径で検索できている．都市部に関してはベースライン 1 が最も狭い半径となったが，最近傍 POI を利用する提案手法と有意差検定を行ったところ有意な差があるとはいえない結果となった．固定値を用いるベースライン手法（B1）では狭い半径から拡大していく手法であるため，応答時間は長いが範囲設定はできるだけ狭く設定される．そのため，ベースラインと提案手法で有意差がないということから，提案手法の半径設定の誤差はベースラインと同程度に狭く設定できていると考えられる．

以上の結果から，提案手法は多様性が同程度であるベースライン手法より高速に検索でき，半径設定の誤差はベースラインと同等であると考えられる．

4.4 定性評価

提案手法では，ユーザが検索する位置によって検索範囲を変えることにより高速に検索を行っている．そこで，定性的な評価としてユーザの検索位置の違いによって半径設定が変わっていることを確認する．東京都における $Tw_{min} = 30$，$R_{max} = 50\,\mathrm{km}$ の場合の最近傍 POI を利用する方法での半径設定の結果の例を，都市部と郊外それぞれ 10 件ずつ**表 5** に示す．検索位置（緯度，経度）は 4.2 節で説明した評価用の位置，市区町村名はその緯度経度が所属する市区町村名，正解半径は 4.1 節で説明した半径設定の正解データ，検索半径は最近傍 POI を利用する提案手法で設定された半径を示す．都市部と郊外では都市部の方が短い半径が設定されている．また，各エリア内でも場所ごとに半径が変わっている．

表 5 半径設定の結果の例
Table 5 Examples of search radius.

エリア	検索位置（緯度，経度）	市区町村名	正解半径	検索半径	再検索回数
都市部	35.750448, 139.738873	北区	500	500	0
都市部	35.722579, 139.584736	練馬区	800	1,000	1
都市部	35.652974, 139.725007	港区	500	500	0
都市部	35.648081, 139.702232	渋谷区	500	500	0
都市部	35.737873, 139.654523	練馬区	500	500	0
都市部	35.694859, 139.706162	新宿区	500	500	0
都市部	35.634137, 139.707142	目黒区	700	700	0
都市部	35.708468, 139.753603	文京区	500	500	0
都市部	35.683156, 139.825862	江東区	1,600	2,000	1
都市部	35.681546, 139.766638	千代田区	500	500	0
郊外	35.717895, 139.567457	武蔵野市	1,300	1,300	0
郊外	35.741159, 139.382659	武蔵村山市	2,900	4,800	1
郊外	35.701759, 139.561015	三鷹市	500	500	0
郊外	35.656621, 139.344087	八王子市	500	500	0
郊外	35.673108, 139.472345	府中市	700	1,000	1
郊外	35.703432, 139.579461	武蔵野市	500	500	0
郊外	35.726066, 139.389948	立川市	2,300	2,400	0
郊外	35.716079, 139.418594	立川市	500	500	0
郊外	35.713019, 139.359581	昭島市	1,800	2,000	2
郊外	35.563328, 139.463329	町田市	2,100	4,000	3

表 5 の最下行（町田市）の事例で再検索回数が多い．事前検索を行った 2016 年 4 月 2 日の検索位置付近のツイートを確認したところ，玉川大学で 2016 年 4 月 2 日に入学式があったことを示すツイートがあった．また，大学のホームページ[*7]でも入学式があったことが記載されていた．評価におけるユーザ検索時（4 月 3 日）には特段イベントはないため，事前検索で検索結果件数に達する半径が狭くなっていたため再検索回数が多かったと考えられる．このような事前検索時にイベントなどがあり投稿数が多く，ユーザ検索時と投稿数の大きく変化する場合は提案手法が有効ではない場合がある．

ここで，どのような場合に提案手法が有効でないか定性的に考察する．検索対象となる場所は以下の 2 パターンが考えられる．

- 観光地や商業施設など日常的に投稿される場所
- 住宅街などふだんは投稿が少ない場所

前者ではイベントなどがなくても日常的に投稿があるため，事前検索の半径が短くても一定の投稿数が得られ，結果として再検索回数は少なくなると考えられる．一方，後者について，事前検索時と投稿数が大きく変わらない場合は問題にならないため，以下の場合の提案手法の挙動を考察する．

(1) ふだん投稿が少ない場所で，事前検索時にイベントな

*7 http://www.tamagawa.jp/education/report/detail_10324.html

表 6 周辺の話題を把握することができる投稿の実例
Table 6 Example of tweets which have nearby trending topic.

投稿日時	POI 名称	位置（緯度，経度）	本文
2016-04-03 00:04:34	六本木ヒルズ	35.660262, 139.729547	「暗殺教室～卒業編～」大ヒット御礼舞台挨拶 六本木ヒルズ@菅田くん>で、泣いてる自分の子供を見て爆笑するっていうね (笑) 涼介>そう (笑) それくらい先の話もできる、考えるくらい良い仲になれたかなって思いますね。
2016-04-03 22:26:23	西武ドーム	35.768539, 139.420527	西武ドーム、熱かった～ももクロちゃん、いつも感動と笑顔をありがとうございます 5 人に出逢えてから、涙腺崩壊 (笑) 今日も何度も泣きそうになっちゃったてか、泣きました 世界のももクロ No.1
2016-04-03 17:01:02	中目黒	35.643854, 139.698621	今日はお花見 中目黒から目黒まで歩いて、JYJ の提灯見て来ました いっぱいあって感激 https://t.co/BK6XgD11yO
2016-04-03 16:54:18	高岡古城公園	36.747323, 137.022138	高岡古城公園の桜 、(｀▽´)/ちょっと小雨 #古城公園 #高岡 #takaoka #toyama #富山 #とやま #さくら #sakura #花見 @ 高岡古城公園 https://t.co/4uz91RP5PC
2016-04-03 21:35:39	秋葉原	35.698619, 139.773288	今まさに満開！痛絵馬も合わせて…秋葉原・神田明神で夜桜撮影してきた！ https://t.co/Fj7oel82MY https://t.co/YREl2CsJwx
2016-04-03 00:00:16	姫路城	34.839431, 134.693933	観桜会が話題ですね https://t.co/wICRU10ncX 姫路城と桜が競演 「観桜会」に市民ら 6 万人 https://t.co/4v60xP27PO
2016-04-03 08:14:44	なかもず	34.55607, 135.50579	今御堂筋線運転再開した？なかもず始発やから座れるな
2016-04-03 08:14:53	西梅田駅	34.699642, 135.495802	8 時 8 分運転再開した！西梅田駅向かう途中で御堂筋線の改札あって良かった (*´ω｀*)

どがあり投稿がふだんより多く，ユーザ検索時はイベントなどがない場合

(2) ふだん投稿が少ない場所で，事前検索時にイベントなどがなく，ユーザ検索時にイベントなどがある場合

(1) の場合は，前述の玉川大学のような事象に該当する．(2) の場合は，ふだん投稿が少ない場所で事前検索時にイベントなどがなければ事前検索では広い範囲を検索する半径が保存される．その状態で，ユーザ検索時にイベントなどがあり周辺の投稿数が増えているとすると，事前検索の半径よりも狭い範囲を検索しても十分な検索結果件数が得られると考えられ，再検索回数は少なくなる．そのため，提案手法で再検索回数が増加することで検索の応答が遅くなるのは (1) の条件の場合と考えられる．

次に，検索結果が周辺の話題を把握するという検索意図に適しているかを確認する．**表 6** に周辺の話題を把握することができる実際の投稿の例を示す．映画の舞台挨拶やライブなどのイベント，花見，電車の運行状況など周辺の話題に関する投稿が得られている．なお，本研究では投稿内容によるフィルタリングは行っていないため，今後，文献 [21] などの観光に適した投稿でのフィルタリングを行うことでより適した投稿を検索できると考えられる．

4.5 事前検索コストの評価

提案手法では事前検索を行うためベースライン手法より計算コストがかかる．本節では，事前検索のコストが実用的な範囲で実施可能か検証を行う．検証に利用したマシンは 4.2 節で説明したものと同じで，シングルスレッドで事前検索処理を実行した．**表 7** に事前検索に要する時間を示す．平均事前検索時間は 4.2 節で説明した評価対象の 3 都府県において事前検索を行ったときの施設あたりの所要時間である．総所要時間は平均事前検索時間を元に全国の POI に実施した場合に想定される時間を示している．

表 7 各パラメータで事前検索に要した時間
Table 7 The result time of pre-search of each parameter.

パラメータ (Tw_{min}, R_{max})	平均事前検索時間（秒）	総所要時間（分）
(10, 10km)	0.452	321.15
(10, 50km)	0.680	482.41
(30, 10km)	0.661	468.91
(30, 50km)	1.068	757.96

検索結果件数や最大半径の増加にともない，事前検索に要する時間は増加している．一般的に商用システムではバッチ処理をユーザからのアクセスが少ない夜間の時間帯に行う[*8]．$Tw_{min} = 30$，$R_{max} = 50\,\mathrm{km}$ 以外の場合は，5～8 時間程度で処理が終わっており検証で利用したスペックのマシン 1 台でも実用的に実行可能である．$Tw_{min} = 30$，$R_{max} = 50\,\mathrm{km}$ の場合も，事前検索は複数の位置での検索を同時に実行可能であるためマルチスレッド処理にするなど並列化を行うことで実用的に利用可能である．事前検索に要する時間が増加するのは，Tw_{min} や R_{max} を増やすなど検索範囲を増やす場合と，事前検索に利用する POI を増やした場合と考えられる．なお，検索対象となる POI 関連ツイートが増加した場合は，一定の検索結果数に到達するのが早くなると考えられるため，再検索回数が少なくなり結果的には事前検索に要する時間が短くなると考えられる．

ここで，事前検索に要する時間と提案手法によって短縮される検索時間のコスト対パフォーマンスの観点で考察する．Google ではウェブサイトの応答時間が 1 秒を超えるとユーザの集中が途切れてしまうため，応答時間は 1 秒未満であることが望ましいとされている[*9]．その

*8 https://www.postgresql.jp/events/enterprise3_nttdata.pdf
*9 https://webmaster-ja.googleblog.com/2013/08/making-smartphone-sites-load-fast.html

うえで，ウェブサイトを表示するために必要な処理を 5 つ（DNS Lookup，TCP Connection，HTTP Request and Response，Server Response，Client-side Rendering）に分け，各処理が 200 ms 以内に終わることが望ましいと述べられている[*10]．この中で本研究で対象としている時空間での投稿の検索は Server Response の部分に該当する．そこで，標準的なウェブアプリケーションの構成とモバイル通信環境を想定して検索に利用できる時間を試算し有用性を考察する．Server Response は，データベースへの問合せとデータ転送に分解できる [22]．データ転送はウェブサイトを構成する HTML，Javascript，CSS とデータを合わせて 1 MB [*11]のデータを，LTE 通信（約 50 Mbps [*12]）で転送すると仮定すると，データ転送には 160 ms を要する．そのため，検索（データベースへの問合せ）に利用できる時間は $200 - 160 = 40$ ms となる．ベースライン手法（B1）と提案手法で検索時間の差は 10 ms 程度だが，応答時間を 1 秒未満とするために投稿の検索に利用できる時間が 40 ms 程度しかなく，その中で 10 ms の速度改善を達成しており，利用可能な検索時間に対して 25%性能を改善しており，改善の効果があると考えられる．

4.6　ユーザビリティ評価

本節では本研究で想定する利用シーンでの提案システムの有用性を主観評価実験により評価する．4.3 節の評価結果より，多様性は範囲検索を行う提案手法とベースライン手法の固定値（B1），Shaw らの手法（B2）では同程度であり，k 近傍検索（B3）とは差があった．そのため，提案手法と k 近傍検索（B3）を対象に評価を行う．評価はシステムの有用性を検証している既存研究 [23] を参考に絶対評価および相対評価を実施した．被験者は 20〜30 代の男女 11 名（平均 32 歳，標準偏差 5.34）である．**図 4** に主観評価で利用した検索システムの画面を示す．画面の右側に提案手法および k 近傍検索で検索した上位 30 件の投稿を表示し，左側には各検索結果に対応する POI の位置を地図上にプロットし表示した．被験者には検索方法は伝えず，画面上では「手法 1」，「手法 2」と表示した．また，地図は上部の地図が手法 1 に，下部の地図が手法 2 に対応している．被験者には，以下の状況を想定し検索結果に対して評価をお願いした．

- 休日に外出し地図上の赤いピンの位置にいると仮定
- これからどこかに出かける，あるいは移動するため周辺で今話題になっていることを知りたいという状況

評価は各検索結果が参考になるか絶対評価で 5 段階評価（1：まったく参考にならない，2：あまり参考にならない，3：どちらともいえない，4：やや参考になる，5：非常に参考になる）と，相対評価としてどちらの検索結果が参考になるか 5 段階評価（1：手法 1 がよい，2：やや手法 1 がよい，3：どちらともいえない，4：やや手法 2 がよい，5：手法 2 がよい）で実施した．被験者には 4.2.2 項で述べた評価用の位置（緯度経度）から 10 地点をランダムに選択し，その場所で周辺の投稿を検索したという仮定で図 4 のシステムで検索結果を提示した．

図 5 に絶対評価の結果を示す．絶対評価は k 近傍検索では平均 2.87，標準偏差 1.02，提案手法では平均 3.17，標準偏差 0.98 であった．これらの結果に対して Wilcoxon の符号検定 [24] を行ったところ，$p = 0.013 < 0.05$ となり有意水準 5%で有意な差があった．**図 6** に相対評価の結果を示す．評価の平均値は 3.34，標準偏差 1.2 であった．評価値 3（どちらともいえない）との差を Wilcoxon の符号検定したところ，$p = 0.010 < 0.05$ となり有意水準 5%で有

図 4　主観評価に用いた検索システムのスクリーンショット
Fig. 4　Screenshot of twitter search system for subjective evaluation.

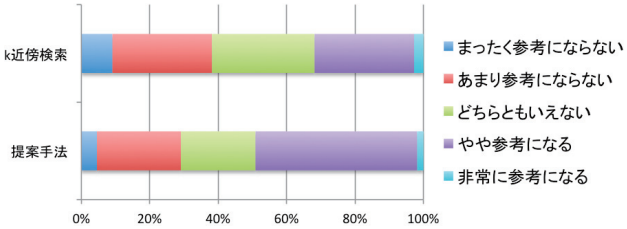

図 5　主観評価の絶対評価の結果
Fig. 5　Result of absolute evaluation of subjective evaluation.

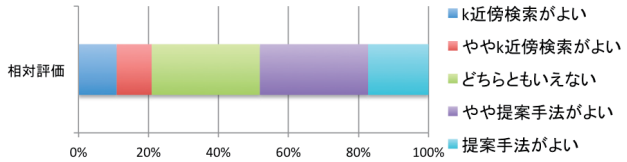

図 6　主観評価の相対評価の結果
Fig. 6　Result of relative evaluation of subjective evaluation.

*10 https://developers.google.com/speed/docs/insights/mobile
*11 ここでは https://twitter.com をログインした状態で読み込んだときに，画面を構成するためのデータが HTML 約 100 KB，Javascript が約 500 KB，CSS が約 150 KB，画像ファイルが約 150 KB であったことから試算した．また，投稿文章のデータ容量は，Twitter API を利用してキーワード検索で上位 30 件のツイートを取得した場合に平均して約 150 KB であったため，合計して 1 MB とした．
*12 http://ictr.co.jp/report/20160929.html

意な差があった．そのため，主観評価の結果から提案手法の方が有用性が高いと考えられる．

5. おわりに

本稿では，Twitter を対象に POI に関連する投稿を位置情報（緯度，経度）を検索クエリとして，位置に応じた検索半径の初期値を設定することで，多様性を考慮しながら高速に投稿を検索する手法を提案した．提案手法では k 近傍検索に比べて多様性が高い検索結果を得ることができる．ユーザの位置に基づいて半径を変化させるため，POI の位置であらかじめ検索結果件数を確保できる最小半径を求める事前検索を行った．ユーザが周辺投稿を検索するときは，半径初期値として事前検索して求めた半径を利用する．半径を一律固定値にする方法と投稿密度に基づいて半径を決定する方法，k 近傍検索をベースラインとして評価を行った．定量評価により k 近傍検索より範囲検索の方が多様性が高いことを検証した．そのうえで，提案手法が多様性を考慮した手法において高速に検索できることを確認した．提案手法ではベースライン手法より約 10.1%〜29.0%高速に検索できる．また，主観評価実験により想定する利用シーンでの有用性を確認した．なお，提案手法の一部は 2012 年 8 月から 2017 年 6 月まで実サービスで利用していた [25]．

今後の課題は，事前検索からユーザ検索までの時間間隔による効率性への影響の調査や，キーワードのインデックスとの統合，事前検索の位置として POI ではなく実サービスログから頻繁に検索される場所を抽出することでより効率的に事前検索が行うことなどが考えられる．

参考文献

[1] Watanabe, K., Ochi, M., Okabe, M. and Onai, R.: Jasmine: A Real-time Local-event Detection System Based on Geolocation Information Propagated to Microblogs, *Proc. 20th ACM CIKM*, pp.2541–2544 (2011).

[2] 落合桂一，鳥居大祐：時間変化する特徴語によるマイクロブログ地名曖昧性解消，情報処理学会論文誌データベース，Vol.7, No.2, pp.51–60 (2014).

[3] 土屋　圭，豊田正史，喜連川優：マイクロブログを用いた鉄道の運行トラブル状況抽出に関する一検討，情報処理学会研究報告情報学基礎研究会報告，Vol.2013, No.31, pp.1–6 (2013).

[4] Nepomnyachiy, S., Gelley, B., Jiang, W. and Minkus, T.: What, where, and when: Keyword search with spatio-temporal ranges, *GIR '14*, Article No.2, ACM (2014).

[5] Bauer, S., Radlinski, F. and White, R.W.: Where Can I Buy a Boulder?: Searching for Offline Retail Locations, *Proc. 25th International Conference on World Wide Web*, *WWW '16*, pp.1225–1235 (2016).

[6] Dhanabal, S. and Chandramathi, S.: A Review of various k-Nearest Neighbor Query Processing Techniques, *Computer Applications*, Vol.31, No.7, pp.14–22 (2011).

[7] Chen, L., Cong, G., Jensen, C.S. and Wu, D.: Spatial keyword query processing: An experimental evaluation, *Proc. 39th International Conference on Very Large Data Bases*, *PVLDB'13*, pp.217–228, VLDB Endowment (2013).

[8] Roussopoulos, N., Kelley, S. and Vincent, F.: Nearest neighbor queries, *ACM Sigmod Record*, Vol.24, No.2, pp.71–79, ACM (1995).

[9] Porkaew, K., Lazaridis, I. and Mehrotra, S.: Querying mobile objects in spatio-temporal databases, *International Symposium on Spatial and Temporal Databases*, pp.59–78, Springer (2001).

[10] Magdy, A., Mokbel, M.F., Elnikety, S., Nath, S. and He, Y.: Mercury: A Memory-Constrained Spatio-temporal Real-time Search on Microblogs, *Proc. ICDE 2014*, *ICDE '14*, pp.172–183 (2014).

[11] Shaw, B., Shea, J., Sinha, S. and Hogue, A.: Learning to rank for spatiotemporal search, *WSDM '13*, pp.717–726, ACM (2013).

[12] 河野浩之，川原　稔：Web 検索におけるテキストマイニング，人工知能学会誌，Vol.16, No.2, pp.212–218 (2001).

[13] Cheng, Z., Caverlee, J. and Lee, K.: You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users, *ACM CIKM '10*, pp.759–768 (2010).

[14] Kitamoto, A.: Toponym-based Geotagging and Disambiguation for Social Media on Earthquake and Weather Events, *10th International Conference on ISCRAM 2013*.

[15] Kim, Y., Han, J. and Yuan, C.: TOPTRAC: Topical Trajectory Pattern Mining, *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *KDD '15*, pp.587–596 (2015).

[16] Amitay, E., Har'El, N., Sivan, R. and Soffer, A.: Web-a-where: Geotagging Web Content, *Proc. ACM SIGIR'04*, pp.273–280 (2004).

[17] 杉谷卓哉，白川真澄，原　隆浩，西尾章治郎：教師あり機械学習を用いたツイート投稿時のユーザ位置推定手法，情報処理学会研究報告 DBS，Vol.158 (2013).

[18] 伊藤嘉博，中村嘉隆，高橋　修，白石　陽：都市センシングデータに基づく経路推薦のための空間補間手法の提案と実装，情報処理学会論文誌，Vol.54, No.2, pp.740–752 (2013).

[19] 張　長平：地理情報システムを用いた空間データ分析，古今書院 (2009).

[20] 宮西大樹，関　和広，上原邦昭：マイクロブログ検索のための時間情報と非時間情報を統合したクエリ拡張，情報処理学会論文誌，Vol.54, No.4, pp.1655–1666 (2013).

[21] 渡邉小百合，吉野　孝：観光地名なしツイートからの観光地に関する感想の抽出手法，情報処理学会論文誌，Vol.59, No.1, pp.43–51 (2018).

[22] 石倉直弥，内海宏律，齋藤邦夫，手塚　大：サーバサイドレンダリングを用いたコンポーネントフレームワークの負荷分散，日立ソリューションズ東日本技報 = Hitachi Solutions East Japan Technical Report，No.22, pp.25–29 (2016), 入手先 ⟨http://ci.nii.ac.jp/naid/40021025645/⟩.

[23] 藤堂祐樹，西村良太，山本一公，中川聖一：複数の対話エージェントを用いた雑談指向の音声対話システム，電子情報通信学会論文誌 D，Vol.99, No.2, pp.188–200 (2016).

[24] Wilcoxon, F.: Individual comparisons by ranking methods, *Biometrics Bulletin*, Vol.1, No.6, pp.80–83 (1945).

[25] 鳥居大祐，赤塚　隼，落合桂一，角野公亮：生活密着情報を提供するリアルタイム検索サービスの開発，NTT DOCOMO テクニカルジャーナル，Vol.20, No.4, pp.12–17 (2013).

落合 桂一　（正会員）

2006 年千葉大学工学部情報画像工学科卒業．2008 年同大学院博士前期課程修了．同年株式会社 NTT ドコモ入社．2017 年東京大学大学院工学系研究科技術経営戦略学専攻博士後期課程修了．博士（工学）．SNS や位置情報，行動ログおよびヘルスケアデータ解析の研究開発に従事．ACM，日本データベース学会，人工知能学会会員．

鳥居 大祐　（正会員）

2001 年京都大学工学部情報学科卒業．2006 年同大学院社会情報学専攻にて博士（情報学）を取得．同年より株式会社 NTT ドコモに所属し，検索システムや位置情報解析に取り組む．2014 年 2 月より株式会社みらい翻訳にて機械翻訳を中心とした自然言語処理の技術開発に従事．2018 年 1 月より同 CTO．言語処理学会会員．

深澤 佑介　（正会員）

2002 年東京大学工学部卒業．2004 年東京大学大学院工学系研究科修士課程修了．同年株式会社 NTT ドコモ入社．2011 年東京大学大学院工学系研究科博士後期課程修了．同年 10 月より東京大学人工物工学研究センターにて協力研究員兼任．IEEE，人工知能学会各会員．博士（工学）．

松尾 豊　（正会員）

1997 年東京大学工学部卒業．2002 年同大学院博士課程修了．博士（工学）．産業技術総合研究所，スタンフォード大学を経て，2007 年東京大学大学院工学系研究科技術経営戦略学専攻准教授．2012 年人工知能学会理事・編集委員長，2014 年より倫理委員長．人工知能学会論文賞，情報処理学会長尾真記念特別賞，ドコモモバイルサイエンス賞等受賞．専門は，Web 工学，Deep Learning，人工知能．