Fast Spatial Twitter Search Method Using Location Adaptive Range Query

Keiichi Ochiai*†, Daisuke Torii*, Yusuke Fukazawa* and Yutaka Matsuo†
*NTT DOCOMO, Yokosuka, Kanagawa Japan. †The University of Tokyo
Email: {ochiaike, toriid, fukazawayu}@nttdocomo.com, matsuo@weblab.t.u-tokyo.ac.jp

***Abstract*—We propose a fast method for searching nearby tweets using location adaptive range query under constraint of the minimum number of tweets. To efficiently search tweets based on the user's location, we need to know the the minimum radius which contains a certain number of tweets. For this purpose, the Point-of-interests (POI) are used as frequently searched locations. Nearby tweets are searched based on POI-associated radius which is explored in advance as the initial range. Our evaluation on one million tweets shows that the proposed method outperforms the baselines using tweet density or fixed radius.**

I. Introduction

With the widespread of smartphones, we have a greater opportunity to search local information on the go. In local search, a user can search static information such as name, address and business hours of Point-of-Interest (POI) etc. On the other hand, social networking services such as Twitter and Facebook have become popular. It is possible to find the trend of real world such as events and POIs which have received a lot of attention[1], [2] by analyzing user's posts. Thus, we can get more instantaneous information from POI-related posts which are referred to POI than local search. 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 POI name or event name. In this research, we suppose that the user would like to search real-time information around the user to determine where to go next. In this situation, following conditions are desired to the search result.

1) Tweets referred to closer POI are preferred. There is supposed to be an allowable (maximum) distance.
2) A certain number of tweets are needed to grasp the nearby topic. By offering a certain number of POI-related tweets, the user can know the hot topic nearby.
3) The search result are sorted by timestamp. Because the merit of using tweets as local information source is high frequency of update, it is useful to sort tweets based on timestamp.

Based on the above requirements, there is a problem in the geographic range setting of the search. Our system needs to retrieve from a smaller region in areas where tweets are dense (e.g. urban area) than we do in areas where tweets are sparse (e.g. suburbs area) to response quickly. Most relevant to our work is the research of Shaw et al. [3] to search POIs for check-in in foursquare service. Shaw et al. proposed a POI density based method to provide approximately 150 POIs for foursquare user to check-in. The problem setting is very similar to our research. However, POIs are relatively static data compared to Twitter data. Therefore, we need to consider temporal information about retrieved data.

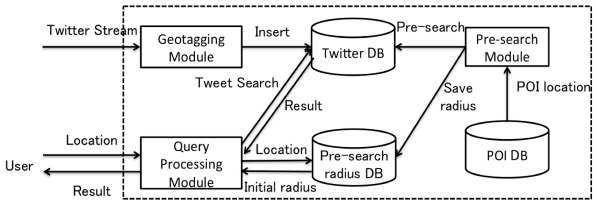

Fig. 1. Overview of the proposed method

In this study, we propose a fast method for retrieving nearby tweets using location adaptive range query. To efficiently search tweets associated with POI based on the user's location, the minimum radius to get a certain number of tweets is explored in advance (we call pre-search) at frequently searched locations. For this purpose, the Point-of-interests (POI) are used as frequently searched locations. Pre-search is executed regularly. Nearby tweets are retrieved based on POI-associated radius which is inquired in advance as initial range.

II. Proposed Method

Figure 1 shows the overview of the proposed method. The proposed system has three main module, namely, geotagging, pre-search and query processing. In geotagging module, each tweet is associated with POI by using toponym disambiguation method [4] and inserted into Twitter DB. Pre-search module investigates minimum radius which we can obtain enough tweets (the minimum number of tweet threshold Tw_{min}), and then saves radius into pre-search radius DB associated with the location of POI. When user search nearby tweets, query processing module receives the user's location and queries pre-search radius DB to get initial radius. Then, query processing module retrieves nearby tweet by using the user's location and pre-searched initial radius.

A. Pre-search

In pre-search, we explore the minimum radius which we can get nearby tweets more than Tw_{min}. At this time, we use the locations (latitude, longitude) of each POIs as queries of frequently searched locations for nearby tweet search, because POI tend to exist where people often visit. Pre-search is executed periodically (e.g. once a day). Algorithm 1 shows pre-search procedure. Here R is radius for nearby tweet search, R_{init}^{pre} is initial radius for pre-search, R_{extend} is incremental radius and R_{max} is the maximum radius for nearby tweet

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** $|TwList| \leq Tw_{min}$ or $R < R_{max}$ **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** $|TwList| \leq Tw_{min}$ or $R < R_{max}$ **do**
5: $R \leftarrow R \times 2$
6: $TwList \leftarrow$ search tweet at (lat, lng) and radius R
7: **end while**
8: **return** $TwList$

search. In our system, because the purpose of our method is providing closer tweet which is preferred, we suppose that the user have maximum allowable travel distance.

B. Query Processing

We propose two method to search nearby tweets using pre-searched radius. That is, 1) Nearest POI method and 2) Grid based method. Both methods have almost common procedure described in Algorithm 2. Difference is how to set initial radius R_{init} (Line 2) which is explained in the following.

Nearest POI Method: First, nearest POI of the user's current location is retrieved, and we get initial radius R_{init} which is associated with nearest POI. Then, nearby tweet is searched at the user's current location and radius R_{init}.

Grid based Method: In grid based method, in addition to pre-search as mentioned above, we calculate mean radius about POIs which are contained within the grid. The mean radius is saved in pre-search radius DB associated with grid code. In this research, we use all types of grid defined by Japanese Statistics Bureau[5] from Quarter Grid Square (250 meters on a side) to Primary Area Partition (50 kilometers on a side). Each grid have different grid code. When the user retrieve nearby tweets, the user's current location is converted into grid code and grid associated initial radius is obtained.

III. Evaluation

In this section, we describe the experimental evaluation. We compare the proposed method with naive method which uses fixed initial search radius regardless of the location (baseline 1, B1) and the method of Shaw et al. [3] (baseline 2, B2). We suppose that all tweets can be regarded as relevant because all tweets contain POI name. Therefore, we use mean response time (MRT) and mean absolute error (MAE) of radius as the metrics of the evaluation, not precision-recall which is commonly used. Each parameters are set as follows: R_{init}^{pre} =500m, R_{extend} =100m, R_{max} =10km, 50km, Tw_{min} =10, 30, R_{init} =500m. We use all combinations of these parameters for evaluation. The Twitter data were collected from 2nd to 3rd in April, 2016. The total number of POI-associated tweet is 925,418. We select Tokyo, Kanagawa and Osaka as evaluation areas and use the locations of randomly sampled 1,000 geo-tagged tweets as query locations to reflect the bias of actual people's location. We divided each prefecture into the urban and the suburbs area. We define urban as the area where the population is over half a million (government-designated city) and the suburbs as other areas. Tweets of April 2, 2016 were used for pre-search and tweets of April 3, 2016 for evaluation. Ground truth of radius is computed by linearly extending tweet search radius at interval of 100m. We use the POI database of Japanese Sightseeing spot obtained from "Local Guide" provided by NTT DOCOMO. The total number of POI is about 30,000. The experiments were conducted on Intel Xeon workstation with 2.60GHz CPU, 64GB memory. We use PostgreSQL 9.4.1, PostGIS 2.1 and enable spatial index.

Table I shows the experimental result of MRT and MAE, bracket indicates standard deviation. The bold style means the best result. †and * indicates statistical significance at the 0.99 and 0.95 level with respect to both B1 and B2. The MRT of nearest POI method is the shortest in both urban and suburbs area. For MAE of radius in urban area, nearest POI method is worse than B1 but there is no statistical significance. In addition, in suburbs area, there is no statistical significance between nearest POI method and both baselines. Therefore, we can conclude that the proposed method is faster than baseline and radius error is at the same level of baselines.

TABLE I
THE RESULT OF MEAN RESPONSE TIME AND MAE OF RADIUS.

	Mean Response Time (in ms)		MAE of Radius (m)	
Method	Urban	Suburbs	Urban	Suburbs
Proposed (Nearest POI)	**85.9**†(67.0)	**54.2** †(14.0)	105.9	427.2
Proposed (Grid based)	99.4* (67.6)	66.8†(12.8)	118.4	455.9
Fixed initial radius (B1)	99.9 (65.8)	67.5 (12.0)	101.5	435.9
Shaw et al.[3] (B2)	100.4 (66.9)	67.6 (12.0)	106.4	435.9

IV. Conclusion

We have proposed a fast method for searching nearby tweets using location adaptive range query. By pre-searching to index appropriate search radius, we can efficiently retrieve nearby tweets. The evaluation results show that the proposed method is faster than the baseline method. The part of the proposed method is used for actual service at NTT DOCOMO[4].

Acknowledgment

We would like to thank Fatina Putri for proof reading.

References

[1] Jaime Teevan, Daniel Ramage and Merredith Ringel Morris, #TwitterSearch: A Comparison of Microblog Search and Web Search, In Proc. of ACM WSDM '11 pp.35-44, (2011).
[2] S. Nepomnyachiy, B. Gelley, W. Jiang and T. Minkus: What, where, and when: keyword search with spatio-temporal ranges. Proc. GIR '14, Article No. 2, ACM, (2014).
[3] B. Shaw, J. Shea, S. Sinha and A. Hogue: Learning to rank for spatiotemporal search. In Proc. of ACM WSDM '13, pp.717-726, (2013).
[4] Daisuke Torii, Hayato Akatsuka, Keiichi Ochiai and Kousuke Kadono, "Development of Real-time Search Services Offering Daily-life Information", NTT DOCOMO Technical Journal, Vol.14, No.4, pp.10-16, 2013.
[5] Japanese Statistics Bureau, http://www.stat.go.jp/english/data/mesh/05.htm