IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 23, 2020, accepted December 2, 2020, date of publication December 7, 2020,
date of current version December 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042813

An Entity-Based Fine-Grained Geolocalization of User Generated Short Text

YONGJUN LI[1], **(Member, IEEE), WENLI JI**[2], **YAO DENG**[2], **AND XING GAO**[2]
[1]School of Computer, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

Corresponding author: Yongjun Li (lyj@nwpu.edu.cn)

This work was supported in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6063.

ABSTRACT Recently, the fine-grained geolocalization of user-generated short text (UGST), which can benefit many location-based applications, has been attracting the attention of academica. The semantic information in UGST is seldom introduced in most existing work, which reduces the effectiveness of existing methods. To address this issue, we propose an entity-based fine-grained geolocalization of UGST, which consists of following steps. (1) We employ location-based social network to model the coupling between entities and locations, which can introduce much semantic information. (2) We extract entities from non-geotagged UGST, and discards this UGST if it has not location-related entities. Otherwise, (3) we utilize the built coupling model to rank the candidate locations for this UGST, and then select top n locations as the result. The experiments demonstrate that our method shows marked improvement on $Accuracy@1km$ and *average error distance* compared to the state-of-the-art FRV, WMV and LW methods.

INDEX TERMS Entity-based method, fine-grained location, geolocalization, location-based social network, user generated short text.

I. INTRODUCTION

With the bloom of social sites, such as Twitter and WeChat, millions of user-generated short texts (UGSTs) are appearing every day. These UGSTs cover almost all aspects of users, including daily routines, news stories, political opinions [1], etc. The value of UGSTs has been attracting considerable attention. Furthermore, UGSTs with fine-grained geolocation [2] can benefit many location-based applications, such as smart health [3], emergency analysis [4], [5], event detection [6], and user identification [7]–[9].

Most operators of social sites have ascertained the value of UGSTs with fine-grained location and provided the geotagging function to their users. However, due to privacy or other special reasons, few users have adopted the geotagging feature. Existing work has illustrated that exceedingly few UGSTs are geocoded with fine-grained location [9]–[12]. For example, of over 1 billion tweets, only 0.58% are geocoded [12]. In this situation, it would be very difficult to fully exploit the value of UGSTs and seize the business opportunities. Therefore, the geolocalization of UGSTs has become a problem that needs to be addressed. We focus on this issue, which differs from existing methods of coarse-grained geolocalization. Generally, these coarse-grained geolocalization methods work by linking the UGSTs to city- or time zone-level locations [10], [11], which are less useful for applications than are fine-grained locations.

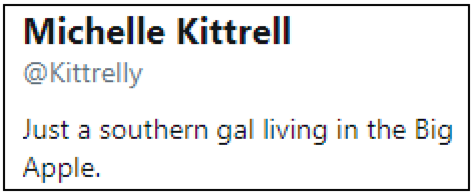
In existing fine-grained geolocalization work, Kinsella *et al.* [13] created the language models of locations using coordinates extracted from geotagged tweets and then employed the content similarity to geolocalize the non-geotagged tweets. Paraskevopoulos and Palpanas [14] considered the time-evolution characteristics to improve the above method. Gonzalez Paule *et al.* [2] presented a solution for the fine-grained geolocalization of tweets, which utilizes a ranking algorithm combined with majority voting of tweets weighted based on the source credibility. Chong and Lim [10], [11] leveraged three types of information from locations, users and peers to rank the fine-grained geolocalization. Gao *et al.* [15] utilized the weight probability model to geolocalize UGST.

Existing work heavily relied on the GPS/human-annotated UGST. However, as mentioned above, when users are less willing to actively geocode the UGSTs [12], fine-grained geolocalization becomes a very challenging issue. To address this problem, Lee *et al.* [12] introduced Foursquare as a

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

source for building the probabilistic models for locations using location-coupled words in tweets, and then proposed a Filtering-Ranking-Validating method for tweet location prediction. Intuitively, we believe that *entity* contains more accurate location information than *word*. We take the instance shown in Figure 1 to illustrate this issue. In tweet *Just a southern gal living in the Big Apple*, entity *Big Apple* clearly refers to the location *New York*, and it contain more location information than word *big* or *apple*.



Michelle Kittrell
@Kittrelly

Just a southern gal living in the Big Apple.

FIGURE 1. An instance of tweet.

According to the above analysis, we propose an entity-based Fine-grained Geolocalization of user-generated Short Text based on a Location-based social network (LBSN), which is abbreviated FGST-L. We first build the probability model for location using the location-coupled entity. Based on the built model, we geolocalize the non-geotagged UGST as the following steps. We identify the entities in the non-geotagged UGST. For an UGST with location-related information, we rank the candidate locations for it, and then we select the top n ($n \geq 1$) locations as the result. For an UGST without any location-related information, we believe that its location is unpredictable. Our contributions are summarized as follows.

(1) We propose an entity-based solution for the fine-grained geolocalization of UGST, which introduces more semantic information to improve the method performance.

(2) We employ the location-based social network to build the coupled model of entity and location, which can introduce more semantic information than existing work. To the best of our knowledge, this is earlier work towards exploiting the coupled relation between entity and location.

(3) We present a novel entity-based method to filter out UGST without any location-related information, which has better filtering effect and eliminates interference in earlier stage.

(4) We conduct the experiment on three ground-truth datasets, and the results illustrate the superiority of FGST-L over the state-of-the-art methods.

It should be mentioned that we had presented the main idea of FGST-L at RecNLP 2019.[1] According to the comments from workshop attendees, we revised and extended the presentation into the mature work, and wrote this paper. The rest of this paper is organized as follows. We introduce the related work in Section 2. Section 3 first describes the preliminary concept, then provides the problem formulation and details our proposed method. Section 4 then shows the experiments on three ground-truth datasets and the result analysis. In Section 5, we conclude this paper and discuss the future work.

II. RELATED WORK

Recently, the geolocalization of UGST has been attracting significant attention from many scholars. The related work can be primarily categorized into two categories. One is coarse-grained geolocalization, which focuses on predicting the *country*, *state*, and *city* of each UGST or its user. The other is fine-grained geolocalization, which focuses on predicting the *street* or *place of interest* of an UGST. In this section, we discuss these two categories of related work.

A. COARSE-GRAINED GEOLOCALIZATION OF UGST

For these methods on coarse-grained geolocalization, the basic idea is building probability models for each *country*, *state* or *city* using region-specific terms and then predicting the location of the UGST or its user according to location-related words in UGST or UGSTs of a user. Concretely, Cheng *et al.* [16] presented a solution for predicting the *city* of a Twitter user. After building the probability model for every city using tweets associated with that city, the probabilities of a user being located in every city are estimated and ranked, and then the city with the highest probability is selected as the city of that user. Hecht *et al.* [17] utilized the selected region-specific terms to build the probability model and then employed a multinomial naive Bayes model to predict the *country* and/or *state* of the Twitter user. Mahmud *et al.* [18] built a set of classifiers for predicting the *home* of a Twitter user and then created an ensemble of these classifiers to improve the accuracy. Huang and Carley [19] integrated the text and user profile into a single model using a convolutional neural network to predict a Twitter user's *country*- or *city*-level location based on the information in a single tweet. Kinsella *et al.* [13] used the coordinates extracted from geotagged tweets to create the probability models of locations at multiple granularities, ranging from the zip code to the country level, and then predicted the location of a single tweet. Ebrahimi *et al.* [20] first proposed a solution for categorizing celebrities as *local* or *global* and then used *local* celebrities as location indicators. A label propagation algorithm was employed over the social network for geolocalization at the *city* level. Finally, a text-based method was integrated into the network-based proposed approach to improve inference accuracy. The difference between our work and these coarse-grained geolocalization methods is that we predict the fine-grained location of a given UGST, such as a street or special restaurant.

B. FINE-GRAINED GEOLOCALIZATION OF UGST

Most existing work on fine-grained geolocalization focuses on predicting the location of each UGST at the *place of interest*-level. Similarly, their fundamental ideas also include

[1] RecNLP was a venue for discussion, and no official proceedings were published. see http://recnlp2019.github.io/

building a probability model for each PoI using PoI-specific terms and then predicting the location of the non-geotagged UGST according to location-related words in the UGST. Li *et al.* [21] predicted the PoI tag of a tweet based on its textual content and time of posting. They considered fine-grained geolocalization as a ranking problem and then ranked a set of candidate PoIs by language and time models. Kinsella *et al.* [13] also created the language models of locations using coordinates extracted from geotagged tweets and then inferred the tweet locations based on content-similarity. Paraskevopoulos and Palpanas [14] improved the method of Kinsella *et al.* [13] by considering time-evolution characteristics in the matching algorithm. Ikawa *et al.* [22] presented a method to learn associations between a location and its relevant keywords from past microblogs and inferred the location where a microblog was generated by using its textual content. Lee *et al.* [12] introduced Foursquare as a source for building the probabilistic models for locations using location-coupled words in tweets and then geocoded the non-geotagged tweets. Li and Sun [23] extracted PoI-level locations mentioned in tweets with temporal awareness. To formulate the PoIs' formal names and their informal abbreviations, they also introduced the crowd wisdom of the Foursquare community into the proposed method. Chong and Lim [10], [11] proposed several models that leverage three types of signals from locations, users and peers to infer the locations of non-geotagged tweets. Gonzalez Paule *et al.* [2] presented a ranking algorithm combined with majority voting for tweets weighted based on source credibility to predict the fine-grained locations of tweets. Whereas most relevant existing methods are based on the probabilistic models for locations using location-coupled *words* in UGSTs, our proposed method attempts to build the probabilistic models for locations using location-coupled *entities* in UGSTs because we intuitively believe that *entities* contain more location information than do *words*.

In addition, Ghaffari *et al.* [24] develop a deep-learning solution for fine-grained home location prediction. Xu *et al.* [25] proposed a deep-learning method for fine-grained location recognition. These two methods have the different goals from our work. Besides, other methods focused on inferring the geographical origins of online contents [12] such as photographs [26], web pages [27] and web search query logs [28].

III. FINE-GRAINED GEOLOCALIZATION OF USER GENERATED SHORT TEXT

We first list some notations which are used in this paper in Table 1, and then formulate the problem of fine-grained geolocalization of user-generated short text. Finally, we detail the proposed approach.

A. PROBLEM FORMULATION

Intuitively, in an UGST, the word group *Big Apple* contains more semantic information of location than does the single word *big* or *apple*. Such word group would be more helpful for the geolocalization of UGST. In FGST-L, we focus more on *word group* than on *word*. For convenience, we call such a *word group* an *entity*.

TABLE 1. Definitions of Notations.

Notations	Definitions
t	a UGST
e	a entity
w	a word
l	a location
L	a set of locations
$T(l)$	a set of UGSTs tied to l

Definition *1:* **Entity.** *An entity is defined as a set of words which represents the name of a subject or object. An entity is further formalized as* $e = \{w_1, w_2, \ldots, w_n\}$*, where* w_i *is the* i^{th} *word of the name.*

An UGST t is further denoted by $t = \{e_1, e_2, \ldots, e_m\}$, where e_i is the i^{th} entity in t. Our goal is to exclusively geolocalize the UGST t in a fine-grained manner based on the entities contained in t.

Problem Formulation *1:* **Fine-Grained Geolocalization of UGST.** *Suppose we are given an UGST* t *and a set of fine-grained candidate locations* $L = \{l_1, l_2, \ldots, l_k\}$*. The task of FGST-L is to select* $n(n \geq 1)$ *locations from* L *as the geolocation of UGST* t*.*

The key issue of the problem 1 is to calculate the probability, $p(l_i|t)$, $\forall l_i \in L$, that the geolocation of t is l_i. After we calculate the probability $p(l|t)$ for each candidate location, we rank these locations according to their probabilities, and then select the top n locations as our results. We will detail this key issue in the following subsection. Clearly, this problem is easily generalized to the coarse-grained geolocalization of UGST.

B. OVERVIEW OF FGST-L

Figure 2 shows the framework of FGST-L, which consists of four key components.

1) **Building the coupled probability model of entity and location**: we employ LBSN, such as Foursquare, as source to build the coupled relationship between entities and locations, which allows us to introduce more semantic information.
2) **Extracting entities in UGST**: we extract the entities in UGST t.
3) **Filtering the UGSTs**: we filter UGSTs without any location-related entities, which are considered as the unpredictable UGSTs.
4) **Ranking the candidate locations**: given an predictable UGST t, we calculate the probability $p(l|t)$ for each candidate location, rank these candidate locations, and select the top $n(n \geq 1)$ locations for t.

We detail four components as follows.

C. BUILDING COUPLED PROBABILITY MODEL OF ENTITY AND LOCATION

Foursquare contains numerous Points of Interest (PoI) and a large amount of tips,[2] which makes it is very helpful for

[2] In Foursquare, a tip is an UGST tied to a PoI.

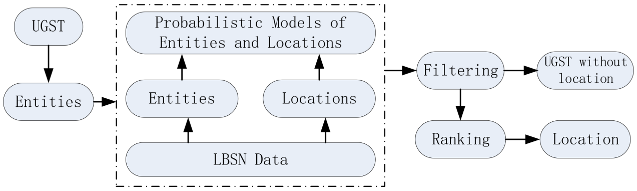


FIGURE 2. Framework of FGST-L. FGST-L includes two stages. One is the stage of pre-training where we employ LBSN to model the coupling between locations and entities. The other is the stage of entity-based geolocalization. The latter consists of three parts: 1) extracting entities in UGST; 2) filtering the UGSTs without any location-related entities; 3) for the remaining UGST, computing the probability of candidate locations based on the built coupling model, and ranking the candidate locations.

building high-quality probability models of locations and entities [12].

We denote the PoIs in Foursquare as $L = \{l_1, l_2, \ldots, l_k\}$. To express the coupling of entities and locations, we build the conditional probability model for every PoI based on its related tips. We assume that the set of tips tied to l_i is $T(l_i) = \{t_1, t_2, \ldots, t_m\}$. Actually, a popular PoI contains more tips, and thus, its model is of higher quality. We assume that entity e occurs $tf(e, t)$ times in UGST t, and $c(e, l)$ times in $T(l_i)$. We compute the probability of entity e occurring in PoI l by the technique of maximum likelihood estimation, as shown in Eq.(1).

$$\begin{aligned} p(e|l) &= \frac{c(e,l)}{\sum_{e_k \in E(l)} c(e_k, l)} \\ c(e,l) &= \sum_{t \in T(l)} tf(e,t) \\ E(l) &= \{e | e \in t, t \in T(l)\} \end{aligned} \tag{1}$$

From Eq.(1), we can easily find that a zero-probability problem, $p(e|l) = 0$, occurs when $c(e, l) = 0$. To address this issue, we further define $p(e|l)$ by the Laplace smoothing method as follows.

$$p(e|l) = \frac{c(e,l)+1}{\sum_{e_k \in E(l)} (c(e_k, l)+1)} \tag{2}$$

Furthermore, in UGST t, some entities maybe have common word(s), which creates difficulties to compute the probability of UGST t being tied to PoI l. Generally, because an UGST is very short, it is relatively rare for words to be shared between entities. We assume that entities in t are independent, and approximate $p(t|l)$ as follows.

$$\begin{aligned} p(t|l) &\approx \prod_{e_i \in t} p(e_i|l) \\ &= \prod_{e_i \in t} \frac{c(e_i,l)+1}{\sum_{e_k \in E(l)} (c(e_k, l)+1)} \end{aligned} \tag{3}$$

D. EXTRACTING ENTITIES IN UGST

In FGST-L, we break each UGST into words, stem them, and remove stop words. After that, an UGST t is denoted by a set of words, $t = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$.

We utilize the Stanford NLP tool [29] to find all possible entities in t based on Microsoft Probase [30]. As a result, we obtain $t = \{e_1, e_2, \ldots, e_m\}$, where $e_i = \{w_k, w_{k+1}, \ldots, w_l | 1 \le k \le l \le n\}$. The extracted entities are restricted to the repository we selected. We selected Probase for extracting entities because it includes tremendous concept space and concept clusters.

E. FILTERING UGSTs

Geolocalization of UGST heavily depends on the location information it contains. For example, it is very difficult to geolocalize UGST *It is a good day*. Before predicting the geolocation of an UGST, we first determine whether this UGST contains location-related information. We filter the UGSTs without any location indication.

In some situations, the location-related information explicitly occurs in the UGST. For instance, the entity *Northwestern Polytechnical University* is an explicit location indication in the UGST *I am at Northwestern Polytechnical University now*. To express these cases clearly, we define the indicator function I^{ex} for entity $e_i \in t$.

$$\mathrm{I}^{ex}(e_i, L) = \begin{cases} 1, & e_i \in L \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We further express whether UGST t contains explicit location-related entities by Eq.(5).

$$\mathrm{I}^1(t, L) = \bigvee_{e_i \in t} \mathrm{I}^{ex}(e_i, L) \tag{5}$$

In other situations, the location-related information appears implicitly. For instance, the entity *Big Apple* can implicitly represent the location *New York* in UGSTs. To express these cases, we employ the idea of TFIDF to identify local words [12] and define the following equation.

$$f_{tfidf}(e,l) = \frac{c(e,l)+1}{\sum_{e_k \in E(l)} (c(e_k,l)+1)} \times \left[\ln \frac{|L|}{df(e)+1} + 1 \right] \tag{6}$$

where $df(e)$ is the number of locations with entity e.

We consider that entity e is l-related when $f_{tfidf}(e, l) \ge \theta$, where θ is a given threshold. Clearly, if we set θ with a greater value, the number of local entities would become smaller. The indicator function of implicit location-related entities is defined over $f_{tfidf}(e, l)$ as shown in Eq.(7).

$$\mathrm{I}^{im}(e_i, l) = \begin{cases} 1, & f_{tfidf}(e_i, l) \ge \theta \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

We can depict whether UGST t contains implicit location-related entities by Eq.(8).

$$\mathrm{I}^2(t, L) = \bigvee_{l \in L} \bigvee_{e_i \in t} \mathrm{I}^{im}(e_i, l) \tag{8}$$

An indicator function $\mathrm{I}(t, L)$ is defined to indicate whether or not t contains location-related entities.

$$\mathrm{I}(t, L) = \mathrm{I}^1(t, L) \bigvee \mathrm{I}^2(t, L) \tag{9}$$

If $\mathrm{I}(t, L) = 1$, t is considered to contain location-related information. Otherwise, it is filtered out.

F. RANKING CANDIDATE LOCATIONS

We rank the PoIs for each remaining UGST. Given UGST $t = \{e_1, e_2, \ldots, e_m\}$ and the candidate PoIs $L = \{l_1, l_2, \ldots, l_k\}$, we rank the candidate PoIs based on the naive Bayes model. Therefore, the probability that the location of t is l is shown by Eq.(10).

$$\begin{aligned} p(l|t) &\propto p(t|l) \times p(l) \approx \left(\prod_{e_i \in t} p(e_i|l)\right) \times p(l) \\ p(l) &= \frac{N(l)}{\sum_{l_i \in L} N(l_i)} \end{aligned} \tag{10}$$

where $N(l)$ indicates the occurrences of l. Eq.(10) is further defined as Eq.(11)

$$\ln p(l|t) \propto \sum_{e_i \in t} \ln p(e_i|l) + \ln p(l) \tag{11}$$

After calculating the probability $p(l_i|t), \forall l_i \in L$, we rank the candidate locations $\{l_i | 1 \leq i \leq k\}$ according to their probabilities and obtain a ranking list $\{l_{r_1}, l_{r_2}, \ldots, l_{r_k}\}$. The top $n(1 \leq n \leq k)$ locations, $\{l_{r_1}, l_{r_2}, \ldots, l_{r_n}\}$, are selected as the possible geolocations of UGST t.

The overall proposed method can be summarized in two stages. One is the stage of building coupled model of entity and location, as shown in Algorithm 1. The other is the stage of entity-based geolocalization of UGST, as shown in Algorithm 2. The first stage is a pre-training process. We introduce the UGSTs and locations in LBSN to build the coupled model. Obviously, the time complexity depends on $|L|$, $|T(l)|$ and the number of entities in UGST. In the datasets we obtained from real social sites, The vast majority of $|T(l)|$ is less than 50, and almost all the number of entities in UGST is less than 10. In other words, the running time of the first stage mainly depends on the number of locations, $|L|$. From Algorithm 2, we easily reach its time complexity is $O(|L|)$. The state-of-the-art method, FRV [12] has similar idea with FGST-L, and its performance is closest to that of FGST-L. Its computational complexity also depends on $|L|$, $|T(l)|$ and the number of words in UGST. Due to the number of words and the number of entities in UGST having the same order of magnitude, FRV and FGST-L have the same time complexity. In a word, the running time of FGST-L and FRV depends heavily on the number of candidate locations.

Algorithm 1: Outline of Building Coupled Model of Entity and Location

```
Input: $L$, location set of LBSN; $T$, UGST set of LBSN
Output: $\mathcal{P}$, model set of entity and location
$\mathcal{P} \Leftarrow \emptyset$;
for each $l \in L$ do
    $T(l) \Leftarrow$ UGSTs tied to $l$ in $T$ ;
    for each $t \in T(l)$ do
        extract entities from $t$ ;
        $p(t|l) = 0$ ;
        for each $e \in t$ do
            compute $p(e|l)$;
        compute $p(t|l)$;
        $\mathcal{P} \Leftarrow \mathcal{P} \cup \{p(t|l)\}$;
return $\mathcal{P}$;
```

Algorithm 2: Outline of Entity-Based Geolocalization of UGST

```
Input: $\mathcal{P}$, model set of entity and location; $L$, candidate
       locations; $t$, a non-geotagged UGST
Output: $L_t$, locations of $t$
$L_t \Leftarrow \emptyset$;
extract entities from $t$, $t = \{e_1, e_2, \ldots, e_m\}$;
compute $\mathrm{I}^{ex}$ and $\mathrm{I}^1(t, L)$ ;
compute $\mathrm{I}^{im}$ and $\mathrm{I}^2(t, L)$ ;
$\mathrm{I}(t, L) = \mathrm{I}^1(t, L) \bigvee \mathrm{I}^2(t, L)$;
if $I(t, L) = 1$ then
    for each $l \in L$ do
        compute $\ln p(l|t)$ ;
    rank locations based on $\{\ln p(l|t), \forall l \in L\}$;
    $L_t \Leftarrow L_t \cup \{$top $n$ ranked locations$\}$;
return $L_t$;
```

TABLE 2. The cumulative number of tips in TrainingTips over PoIs.

No. of tips	≥ 1	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50
No. of PoIs	74942	9208	4939	3790	2930	2132

IV. EXPERIMENTS

A. DATASETS

We collected the PoIs of *New York* City and the related tips from Foursquare and obtained 74,942 PoIs and 498,722 tips. For convenience, we call this dataset TrainingTips. The number of tips is unevenly distributed over all PoIs, as shown in Table 2. The numbers of tips for approximately 87.7% $[= (1 - \frac{9208}{74942}) \times 100\%]$ of PoIs are less than 10. Only 3,790 PoIs contain more than 30 tips.

To illustrate the generalization of FGST-L, we also collected the UGSTs generated in *New York* City from Twitter and Facebook, and ultimately obtained 19,231 tweets and 6,699 posts. In total, 32.4% of tweets and 16.7% of posts are geocoded by the PoI-level location. In addition to the tips in TrainingTips, we obtained additional tips and their PoIs from Foursquare for evaluation and manually selected 12,000 tips, of which 6,000 tips contain hints about locations and 6,000 tips do not contain any hints about locations. The three datasets are named TW, FB, and FS, respectively.

B. EXPERIMENTAL SETTINGS

In FGST-L, there are three key parameters: the predefined threshold θ, the number of ranked locations selected for t, and the number of tips used for building probability model. For convenience, we denote three parameters by θ, n_{top}, and n_{tip}, respectively. We will discuss the effects of three parameters.

We compare FGST-L with the following similar methods.

- FRV [12]: a filtering-ranking-validating technique for the fine-grained geolocalization of tweets, which is a very similar method to FGST-L. FGST-L is an entity-based method, while FRV is a word-based method.
- LW [10], [11]: a location-indicative weighting scheme for the fine-grained geolocalization of tweets, which assigns more weight to location-indicative words.
- WMV [2]: a weighted majority voting algorithm for the fine-grained geolocalization of tweets, which estimates the location of a tweet by collecting the votes of the geotagged tweets that are similar with that tweet on content.

In experiments, we first use the PoIs and tips in TrainingTips to model the coupling between *entities* and *locations*. Then, we use Algorithm 2 to geolocalize each t in TW, FS and FB, respectively. Based on the geolocalization results, we employ the widely used *Accuracy@1km* and *average error distance (km)* [2] to evaluate all algorithms.

Average Error Distance (km): we only consider the UGSTs that have not been filtered out and compute the distance on Earth between the predicted location and the real coordinates of the UGST.

Accuracy@1km: After filtering, all UGSTs are divided into two categories: UGSTs that have been filtered out and UGSTs that have not been filtered out. We assume that the number of UGSTs that have been filtered out correctly is n^1_{1km}. For UGSTs that have not been filtered out, the number of UGSTs whose predicted location lies within a radius of 1 *km* from the real location is denoted by n^2_{1Km}. *Accuracy@1km* is measured as follows.

$$Accuracy@1km = \frac{n^1_{1Km} + n^2_{1Km}}{n_T} \tag{12}$$

where n_T is the number of all UGSTs for testing.

C. PERFORMANCE OF FGST-L W.r.t θ

Given UGST t, it would be filtered out if $f_{tfidf}(e, l) < \theta$, $\forall e \in t$. In other words, the value of θ determines whether t is filtered out. Therefore, the number of UGSTs that are not filtered out will vary with the value of θ, as shown in Table 3. As the value of θ becomes larger, more UGSTs are filtered out. The UGSTs that are not filtered out are considered predictable.

TABLE 3. Number of UGSTs that are not filtered out w.r.t. θ.

θ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
FS	12,000	10,946	9,387	6,718	3,871	871	151	66
FB	6,699	6,108	5,342	3,771	1,533	261	70	24
TW	19,231	15,478	12,071	7,210	3,020	636	139	26

Naturally, the threshold, θ, exerts a strong influence on the results. We conduct the experiments with the different values of θ to study its effect on the results, where we set $n_{top} = 1$ and $n_{tip} = 20$. The results are shown in Figure 3.

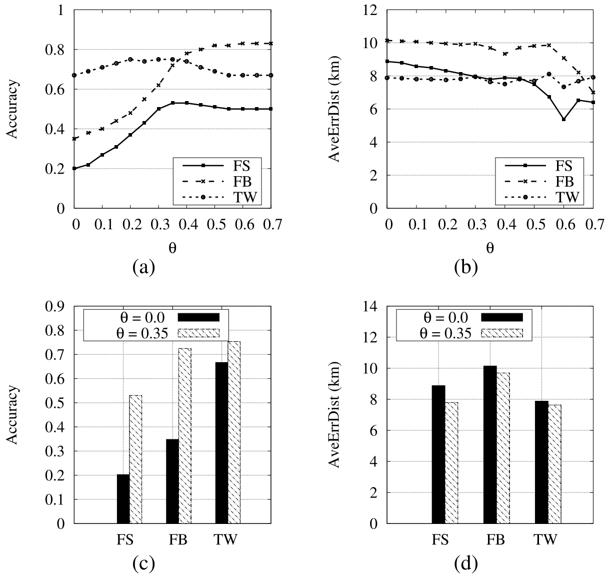


FIGURE 3. Results of FGST-L w.r.t. θ.

Figure 3(a) shows the *Accuracy@1km* w.r.t. θ. For FS and TW, the accuracy of FGST-L first rises and then declines with the increase in θ. Its accuracy exhibits the best performance when $0.3 < \theta < 0.4$. Generally, many UGSTs do not contain location-related information. When θ takes a small value, many UGSTs without location-related entities are mistaken for UGSTs with location information. This reduces the accuracy of FGST-L. Similarly, when θ takes a large value, FGST-L incorrectly filters out the UGSTs with location-related information, which also reduces the accuracy. However, the accuracy curve exhibits a different trend for dataset FB. The curve first rises quickly and then increases only slightly. Because most UGSTs in FB are location-free, an increasing number of location-free UGSTs are correctly filtered out with the increase in θ at the beginning. However, after $\theta > 0.4$, most location-free UGSTs have been filtered out, which slows the increasing trend. The accuracy for the FB dataset always continues to increase, which is primarily due to its severe data skew. When $\theta = 0.7$, FGST-L filters most UGSTs, and its accuracy reaches 83.3%.

The *average error distance* with θ is shown in Figure 3(b). When $0.3 < \theta < 0.4$, this metric is optimal. Although it obtains a minimum value when $\theta > 0.6$, most UGSTs are filtered out in this case, which is not our expectation.

The above results shows that FGST-L has the optimal performance when $0.3 < \theta < 0.4$. Therefore, we can set the value for parameter θ in the interval [0.3, 0.4].

To clearly show the effect of *filtering*, we illustrate the comparison between the unfiltered results ($\theta = 0.0$) and filtered results ($\theta = 0.35$) in Figure 3(c) and Figure 3(d), respectively. Whether for *accuracy* or *average error distance*, the filtered results are significantly better than the unfiltered results. This finding demonstrates that *filtering* is an essential step of FGST-L.

D. PERFORMANCE OF FGST-L W.r.t n_{tip}

Intuitively, if a PoI is tied to more tips, the probability model built for this PoI is of higher quality. Therefore, we conduct experiments to study the effect of n_{tip}, where $n_{top} = 1$ and $\theta = 0.35$. Figure 4 shows the experimental results.

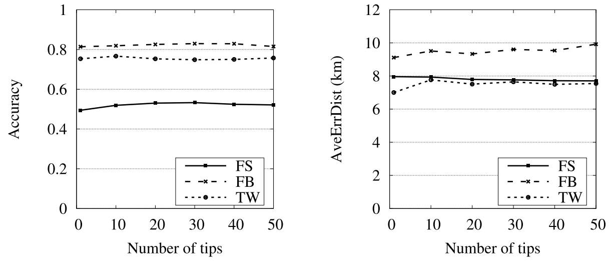


FIGURE 4. Results of FGST-L w.r.t. n_{tip}.

We find that the *accuracy* of FGST-L rises slightly as n_{tip} increases, while the *average error distance* first rises and then declines before $n_{tip} \leq 20$. In other words, the number of tips has some impact on the coupling model, but it is not very significant before $n_{tip} \leq 20$. This result is not in accordance with our intuition. In particular, when $n_{tip} > 20$, it exerts little effect on the probability model. The reasons for this result are as follows: 1) in the experimental datasets, many UGSTs are location-free, which interferes with our prediction, and 2) for the UGSTs tied to a PoI, the entities in 20 UGSTs cover most entities in all UGSTs. As a result, we can build the accurate coupled of entities and PoI with only approximately 20 UGSTs. This could reduce the need for computing resources and help us obtain a much more accurate probability model. Therefore, we recommend that n_{tip} is set to the number of UGSTs covering most entities. In our experiments, we set $n_{tip} = 20$.

E. PERFORMANCE OF FGST-L W.r.t n_{top}

From the above experimental results, we easily find that the best *accuracy* is approximately 80%, as shown in Figure 3 and Figure 4. This could be caused by selecting the top-ranked location as the location of UGST t. Instead, in many cases, the ground-truth location of t is on the $k^{th}(k \geq 2)$ place of the ranking list, not the top-ranked place. Intuitively, if we select the top n_{top} locations as the possible locations of t, the *accuracy* should improve. We conduct experiments to demonstrate the effect of n_{top}, where $n_{tip} = 20$ and $\theta = 0.35$. The results are shown in Figure 5.

From Figure 5(a), we can easily observe that *accuracy* has improved significantly. The detailed percentage of *accuracy* improvement $(= \frac{Acc@Topn_{top} - Acc@Top1}{Acc@Top1} \times 100\%)$ is shown in Table 4. With the increase in n_{top}, *Accuracy* is gradually improving, but the acceleration of the *percentage* gradually decreases. Before $n_{top} = 8$, the improvement is relatively large. These results meet our expectations.

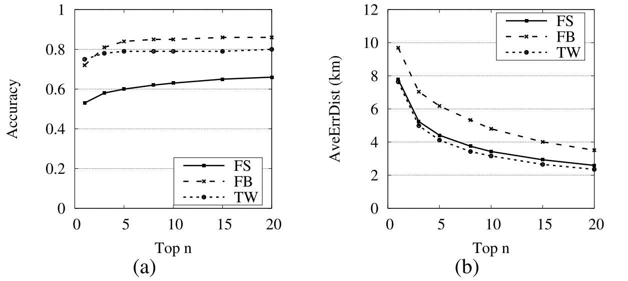


FIGURE 5. Results of FGST-L w.r.t. n_{top}.

TABLE 4. Percentage of *Accuracy* improvement w.r.t. n_{top}.

n_{top}	3	5	8	10	15	20
FS	9.38%	13.91%	17.82%	19.62%	22.85%	24.81%
FB	12.23%	15.52%	17.02%	17.55%	18.48%	19.34%
TW	3.94%	4.65%	4.98%	5.14%	5.43%	5.65%

Among the datasets, the percentage of *accuracy* improvement on FS is the highest, while the percentage on TW is the lowest. We have conducted further analysis and found reasonable explanations for these results. 1) The locations in TW are much more coarse. When $n_{top} = 1$, its *accuracy* is relatively high, as shown in Figure 3(a). As n_{top} increases, the change in the percentage is not obvious. 2) The location granularity of FS is the finest. The candidate locations close to the ground-truth location of t readily interfere with our inference results. Figure 3(a) supports this statement. Obviously, when we select the top n_{top} locations, the *accuracy* for FS improves remarkably. 3) For the FB dataset, only 16.7% of posts are geocoded with fine-grained location. Similar to the reason for TW, the change in the percentage is not apparent after $n_{top} > 3$. In future work, we will extend the datasets and study the relationship between the number of the UGSTs with the fine-grained locations and the accuracy of FGST-L.

Figure 5(b) illustrates the remarkable change in *average error distance*. The percentages of *average error distance* improvement $(= \frac{AveErrDist@Top1 - AveErrDist@Topn_{top}}{AveErrDist@Top1} \times 100\%)$ with n_{top} are detailed in Table 5. When selecting the top n_{top} locations, we take the location closest to the real location as the predicted location. Clearly, as n_{top} gradually becomes larger, the value of *average error distance* becomes smaller. The *average error distance* improves remarkably for the three datasets, particularly before $n_{top} = 10$. As mentioned above, the ratio of *posts* without location information is significantly larger than the ratio of *tips* or *tweets*, so the percentage of *average error distance* improvement for FB is relatively small.

TABLE 5. Percentage of *averageerrordistance* improvement w.r.t. n_{top}.

n_{top}	3	5	8	10	15	20
FS	32.73%	43.40%	51.89%	55.95%	62.43%	66.72%
FB	27.38%	36.21%	45.05%	50.49%	58.65%	63.77%
TW	34.65%	46.02%	55.07%	58.69%	65.26%	69.21%

TABLE 6. Performance of FGST-L with $n_{top} = 1$.

	n_{tip}	FS					FB					TW				
	θ	*10*	*20*	*30*	*40*	*50*	*10*	*20*	*30*	*40*	*50*	*10*	*20*	*30*	*40*	*50*
Accuracy (%)	0.0	21.66	20.26	19.35	18.21	17.06	35.27	34.83	34.65	33.73	32.74	66.59	66.69	66.83	67.17	66.87
	0.1	23.24	27.02	28.13	28.75	29.08	37.55	40.19	41.66	42.47	44.06	68.16	70.92	72.20	73.38	73.28
	0.2	30.15	36.84	40.46	42.82	44.38	41.40	48.21	52.56	55.69	57.47	71.03	74.53	74.05	**74.49**	**74.42**
	0.3	41.20	49.60	52.24	52.44	**52.14**	51.06	61.91	67.35	69.82	75.55	**75.16**	**75.05**	**74.47**	73.32	71.76
	0.4	51.15	**52.72**	**53.27**	**52.49**	51.63	67.24	77.63	80.58	81.48	81.54	74.70	74.23	71.28	69.85	69.02
	0.5	**51.43**	51.08	50.38	50.27	50.03	76.09	81.54	82.74	82.98	83.10	71.21	68.68	67.12	66.90	66.79
	0.6	50.44	50.14	49.95	49.95	49.92	80.68	82.57	82.99	**83.16**	**83.16**	67.92	67.26	66.85	66.76	66.76
	0.7	50.18	50.03	49.91	49.92	49.92	**81.89**	**82.98**	**83.02**	83.16	83.16	66.91	66.84	66.82	66.74	66.74
Ave. Err. Dis.(km)	0.0	8.73	8.87	8.97	9.07	9.16	10.13	10.15	10.12	10.13	10.20	7.93	7.88	7.86	7.78	7.79
	0.1	8.67	8.58	8.62	8.64	8.66	10.08	10.07	10.02	10.01	10.04	7.89	7.80	7.75	7.61	7.61
	0.2	8.42	8.32	8.28	8.26	8.22	10.03	9.96	9.86	9.86	9.90	7.88	7.76	7.76	7.59	7.52
	0.3	8.12	7.97	7.77	7.72	7.71	9.92	9.95	10.02	9.85	10.10	**7.80**	7.95	7.70	7.67	7.83
	0.4	**7.93**	7.88	7.01	6.90	6.10	9.69	9.33	9.61	9.53	10.14	7.98	7.51	7.71	7.75	7.73
	0.5	8.61	7.50	**5.94**	5.53	7.04	**9.48**	9.81	11.11	11.45	12.82	8.11	7.70	8.49	8.64	8.17
	0.6	9.37	**5.38**	6.25	**4.56**	**5.53**	9.70	9.08	**7.62**	**6.33**	**6.33**	8.20	**7.34**	**5.96**	**5.03**	**5.03**
	0.7	7.95	6.41	7.44	6.82	6.82	9.51	**7.00**	8.32	8.92	8.92	8.85	7.92	6.13	6.27	6.09

TABLE 7. Performance of FGST-L with $n_{top} = 5$.

	n_{tip}	FS					FB					TW				
	θ	*10*	*20*	*30*	*40*	*50*	*10*	*20*	*30*	*40*	*50*	*10*	*20*	*30*	*40*	*50*
Accuracy (%)	0.0	37.25	35.13	33.46	31.81	30.14	65.52	65.05	65.22	65.26	64.94	81.46	81.55	81.48	81.41	81.34
	0.1	38.66	41.12	41.27	41.09	40.68	66.89	68.54	69.72	70.99	72.29	81.76	82.97	**83.23**	**83.32**	**82.77**
	0.2	44.70	49.50	51.54	52.65	52.80	69.63	73.92	76.73	79.05	80.27	82.90	**84.18**	82.11	81.09	80.42
	0.3	53.90	**59.13**	**59.65**	**58.46**	**56.74**	75.67	80.11	80.95	81.51	82.90	**84.08**	80.86	78.22	76.13	74.05
	0.4	**60.31**	58.11	55.91	54.24	52.75	82.75	**84.44**	**83.81**	**83.95**	**83.79**	79.48	76.40	72.30	70.56	69.53
	0.5	55.61	52.23	50.78	50.45	50.15	82.89	82.92	83.11	83.11	83.17	73.05	69.17	67.26	66.96	66.82
	0.6	51.81	50.38	49.99	49.97	49.95	**83.25**	82.99	83.10	83.17	83.17	68.42	67.34	66.86	66.76	66.76
	0.7	50.71	50.09	49.93	49.94	49.95	82.86	83.11	83.11	83.17	83.17	67.06	66.86	66.83	66.74	66.74
Ave. Err. Dis.(km)	0.0	5.28	5.46	5.64	5.75	5.88	6.63	6.65	6.62	6.61	6.63	4.39	4.34	4.33	4.35	4.36
	0.1	5.22	5.23	5.37	5.41	5.49	6.59	6.56	6.52	6.47	6.42	4.40	4.32	4.28	4.27	4.25
	0.2	5.01	4.94	5.00	5.00	5.02	6.53	6.42	6.29	6.19	6.12	4.36	4.20	4.13	**4.13**	**4.12**
	0.3	4.67	4.54	4.50	4.50	4.45	6.37	6.37	6.42	6.10	5.91	**4.30**	4.27	4.17	4.15	4.17
	0.4	4.45	4.51	3.81	3.65	3.32	6.11	5.79	5.77	5.60	5.86	4.44	4.08	4.32	4.45	4.35
	0.5	4.83	4.19	**3.26**	3.19	4.27	5.97	6.33	6.78	7.57	9.28	4.37	3.95	4.77	4.54	3.93
	0.6	5.07	**3.11**	3.64	**2.07**	**1.38**	**5.96**	6.55	**5.72**	**4.66**	**4.66**	4.50	3.89	4.25	4.56	4.56
	0.7	**4.09**	3.83	4.39	2.98	1.54	6.19	**5.10**	6.27	7.49	7.49	5.00	**3.66**	**3.87**	5.91	6.05

In summary, we present the comprehensive FGST-L experimental results with $n_{top} = 1$ and $n_{top} = 5$, as shown in Table 6 and Table 7, respectively. Whether $n_{top} = 1$ or $n_{top} = 5$, we easily reach the conclusion that FGST-L performs well when $20 < n_{tip} < 30$ and $0.3 < \theta < 0.4$. When n_{top} takes other values, the performance of FGST-L exhibits a similar pattern.

As n_{top} gets bigger, FGST-L gets better. However, the number of possible locations for t has also become larger, which makes it more difficult for the user to choose one. Therefore, we set $n_{top} \leq 5$.

F. PERFORMANCE OF FGST-L UNDER RELAXED CONDITIONS

In the above experiments, we use the metric *Accuracy@1km* to evaluate the accuracy of FGST-L. If the radius error of the predicted location and the real location is less than 1 km, the inference result is considered correct. Intuitively, the radius error should have a significant impact on the accuracy of FGST-L. We relax the conditions for calculating the metrics of FGST-L to study this issue. Due to limited space, we only demonstrate the results of FS w.r.t. radius error and n_{top}, as shown in Figure 6.

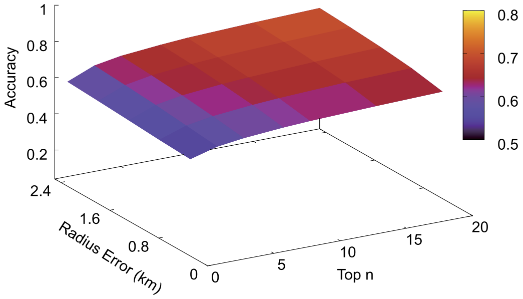


FIGURE 6. Results of FGST-L on FS under relaxed conditions.

From the results, we find that the *accuracy* of FGST-L increases slightly as the radius error becomes larger when n_{top} is given. Similarly, the *accuracy* of FGST-L also increases with the increase in n_{top} when the radius error is given. However, the increase in the latter case is significantly greater than the increase in the former case. This illustrates that the location predicted by FGST-L is fine-grained. To increase the feasibility of FGST-L, we recommend selecting the top n_{top} locations as locations of UGST t.

G. COMPARISON WITH EXISTING WORK

Figure 7 shows the comparison between FGST-L and existing work.

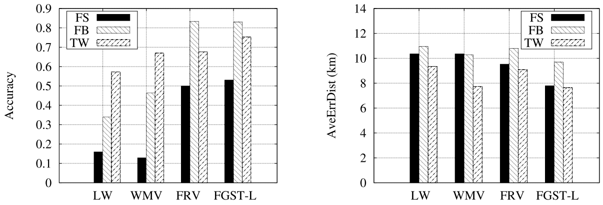


FIGURE 7. Results of FGST-L for FS, FB and TW datasets.

The results demonstrate the effectiveness of FGST-L due to its superiority over the other baseline methods. These results stem from the fact that we 1) build the probability models with entities instead of words, where entities contain more semantic information than do words, and 2) filter out the UGSTs without location-related entities.

FRV exhibits better results than the other baseline methods, WMV and LW, which further indicates that filtering the location-free UGSTs is an effective step that reduces noise interference.

Both FGST-L and FRV exhibit better results on the FB dataset than on the FS and TW datasets. As analyzed above, approximately 83.3% of UGSTs contain few location-related entities in FB. FGST-L and FRV filter them out, which increases the accuracies. This result further demonstrates that determining whether an UGST includes location-related entities is easier than geocoding its PoI. In addition, all methods exhibit inferior accuracies on FS than on FB and TW. This is because FS has more fine-grained location.

We perform a t-test on the results of FGST-L and FRV, and find that there is not significant difference at significance level 0.05. A reasonable explanation is that FRV employ the n-gram model to extract the words, which include most of entities. However, some words that are not entities may be noisy to geolocalization of UGST.

To further validate our method, we remove all location-free UGSTs from three datasets, and rerun four methods. The results are shown in Table 8, where $n_{top} = 5$, $n_{tip} = 20$ and $\theta = 0.35$. Compared with Fig. 7, both FGST-L and FRV also show better accuracy, but their advantages reduce significantly. Four methods show much similar results because they rely on the similar information, location-indicative words/entities. However, in FGST-L or FRV, n-gram model or entity is more location-indicative, so its performance is better. The results on *average error distance (km)* change very slightly.

We present some examples to show the effects of using FGST-L. Table 9 displays three tweets posting at *Joe's Shanghai, New York*. Within each tweet, the entities are italicized. For instance, FGST-L easily recognizes the entity *joe's shanghai* from tweet t_1, and this entity is the name of a Chinese restaurant. Therefore, FGST-L easily geolocalizes t_1. For tweet t_3, FGST-L can not distinguish any entity, and consider t_3 as a location-free tweet. However, for t_2, it is more difficult for FGST-L to geolocalize. Our method can recognizes the entity *soup dumplings*, which is a representative food in Chinese restaurants. In this case, FGST-L will incorrectly geolocalize t_3 with high probability. Similarly, the other three methods are also helpless for tweet t_3.

TABLE 8. Results of FGST-L after removing the location-free UGSTs.

		LW	WMV	FRV	FGST-L
FS	*Acc.@1km*	34.74%	35.89%	39.74%	40.45%
	AveErr.(km)	10.57	9.27	8.33	8.14
FB	*Acc.@1km*	42.25%	50.67%	60.61%	60.60%
	AveErr.(km)	11.34	10.05	10.30	9.47
TW	*Acc.@1km*	45.38%	47.45%	57.13%	61.97%
	AveErr.(km)	8.34	6.75	8.64	7.89

TABLE 9. Three sample tweets posting at *Joe's Shanghai, New York*.

No.	tweet
t_1	so you have checked into *joe's shanghai*.
t_2	*soup dumplings* are amazing.
t_3	come during the day.

V. CONCLUSION AND FUTURE WORK

Recently, the value of a tremendous amount of UGSTs in social networks, particularly the UGSTs tagged with fine-grained locations, has been recognized by increasingly numerous business organizations. However, due to privacy issues or special purposes, most users seldom adopt the geotagging functions provided by social sites. To fully exploit the value of UGSTs, the fine-grained geolocalization of UGSTs has been receiving great attention from academia. Most existing methods are word-based and thus rarely utilize the semantic information about a location. This will degrade the performances of existing approaches. To address this problem, we present an entity-based fine-grained geolocalization of UGST based on LBSN. We introduce LBSNs, such as Foursquare, as sources to tightly couple entities and locations, which capture more semantic information of locations than the word-based methods do. After filtering out the UGSTs without any location-related entities, we rank the candidate locations for each remaining UGST based on the coupling model and then select the top $n (n \geq 1)$ locations as results. The experiments on three ground-truth datasets validate the effectiveness of the proposed method.

To more accurately geolocalize UGSTs, we will extend our method by incorporating more information sources in future work. One extension could be the introduction of UGSTs posted by a user in a LBSN to predict the locations of UGSTs posted by the same user on other social sites. For example, if a user visits one shopping mall at 12 o'clock and simultaneously posts two similar UGSTs on Foursquare and Twitter, then we can accurately predict the location of the UGST on Twitter with the help of the PoI on Foursquare. Another possible extension could be to introduce the location history of a user, which could reduce the search space of candidate locations.

REFERENCES

[1] K. Starbird and L. Palen, "(How) will the revolution be retweeted?: Information diffusion and the 2011 Egyptian uprising," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, New York, NY, USA, 2012, pp. 7–16.

[2] J. D. Gonzalez Paule, Y. Moshfeghi, J. M. Jose, and P. Thakuriah, ''On fine-grained geolocalisation of tweets,'' in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, New York, NY, USA, Oct. 2017, pp. 313–316.

[3] A. Noulas, C. Moffatt, D. Hristova, and B. Gonçalves, ''Foursquare to the rescue: Predicting ambulance calls across geographies,'' in *Proc. Int. Conf. Digit. Health*, New York, NY, USA, 2018, pp. 100–109.

[4] R. McCreadie, C. Macdonald, and I. Ounis, ''EAIMS: Emergency analysis identification and management system,'' in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, 2016, pp. 1101–1104.

[5] T. Sakaki, M. Okazaki, and Y. Matsuo, ''Earthquake shakes Twitter users: Real-time event detection by social sensors,'' in *Proc. 19th Int. Conf. World Wide Web*, New York, NY, USA, 2010, pp. 851–860.

[6] F. Atefeh and W. Khreich, ''A survey of techniques for event detection in Twitter,'' *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015, doi: 10.1111/coin.12017.

[7] Y. Li, Y. Peng, Z. Zhang, H. Yin, and Q. Xu, ''Matching user accounts across social networks based on username and display name,'' *World Wide Web*, vol. 22, no. 3, pp. 1075–1097, May 2019.

[8] X. Gao, W. Ji, Y. Li, Y. Deng, and W. Dong, ''User identification with spatio-temporal awareness across social networks,'' in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2018, pp. 1831–1834.

[9] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Xu, ''Matching user accounts based on user generated content across social networks,'' *Future Gener. Comput. Syst.*, vol. 83, pp. 104–115, Jun. 2018.

[10] W.-H. Chong and E.-P. Lim, ''Tweet geolocation: Leveraging location, user and peer signals,'' in *Proc. ACM Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2017, pp. 1279–1288.

[11] W.-H. Chong and E.-P. Lim, ''Fine-grained geolocation of tweets in temporal proximity,'' *ACM Trans. Inf. Syst.*, vol. 37, p. 17, Jan. 2019.

[12] K. Lee, R. Ganti, M. Srivatsa, and L. Liu, ''When Twitter meets foursquare: Tweet location prediction using foursquare,'' in *Proc. 11th Int. Conf. Mobile Ubiquitous Systems: Comput., Netw. Services*, Brussels, Belgium, 2014, pp. 198–207.

[13] S. Kinsella, V. Murdock, and N. O'Hare, '''I'm eating a sandwich in Glasgow': modeling locations with tweets,'' in *Proc. 3rd Int. Workshop Search Mining User-Generated Contents*, New York, NY, USA, 2011, pp. 61–68.

[14] P. Paraskevopoulos and T. Palpanas, ''Fine-grained geolocalisation of non-geotagged tweets,'' in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, New York, NY, USA, 2015, pp. 105–112.

[15] C. Gao, Y. Li, J. Yang, and W. Dong, ''Fine-grained geolocalization of user-generated short text based on a weight probability model,'' *IEEE Access*, vol. 7, pp. 153579–153591, 2019.

[16] Z. Cheng, J. Caverlee, and K. Lee, ''You are where you Tweet: A content-based approach to geo-locating Twitter users,'' in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2010, pp. 759–768.

[17] B. Hecht, L. Hong, B. Suh, and E. H. Chi, ''Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles,'' in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2011, pp. 237–246.

[18] J. Mahmud, J. Nichols, and C. Drews, ''Home location identification of Twitter users,'' *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 47, 2014.

[19] B. Huang and K. M. Carley, ''On predicting geolocation of tweets using convolutional neural networks,'' in *Social, Cultural, and Behavioral Modeling*, D. Lee, Y.-R. Lin, N. Osgood, and R. Thomson, Eds. Cham, Switzerland: Springer, 2017, pp. 281–291.

[20] M. Ebrahimi, E. ShafieiBavani, R. Wong, and F. Chen, ''Twitter user geolocation by filtering of highly mentioned users,'' *J. Assoc. Inf. Sci. Technol.*, vol. 69, no. 7, pp. 879–889, Jul. 2018, doi: 10.1002/asi.24011.

[21] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson, ''The where in the tweet,'' in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2011, pp. 2473–2476.

[22] Y. Ikawa, M. Enoki, and M. Tatsubori, ''Location inference using microblog messages,'' in *Proc. 21st Int. Conf. Companion World Wide Web (WWW Companion)*, New York, NY, USA, 2012, pp. 687–690.

[23] C. Li and A. Sun, ''Extracting fine-grained location with temporal awareness in tweets: A two-stage approach,'' *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 7, pp. 1652–1670, Jul. 2017.

[24] M. Ghaffari, A. Srinivasan, and X. Liu, ''High-resolution home location prediction from tweets using deep learning with dynamic structure,'' in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, New York, NY, USA, Aug. 2019, p. 540.

[25] C. Xu, J. Li, X. Luo, J. Pei, C. Li, and D. Ji, ''DLocRL: A deep learning pipeline for fine-grained location recognition and linking in tweets,'' in *Proc. World Wide Web Conf. (WWW)*, New York, NY, USA, 2019, p. 3391.

[26] P. Serdyukov, V. Murdock, and R. van Zwol, ''Placing flickr photos on a map,'' in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2009, pp. 484–491.

[27] E. Amitay, N. Har'El, R. Sivan, and A. Soffer, ''Web-a-where: Geotagging Web content,'' in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, 2004, pp. 273–280.

[28] R. Jones, R. Kumar, B. Pang, A. Tomkins, A. Tomkins, and A. Tomkins, '''I know what you did last summer': query logs and user privacy,'' in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2007, pp. 909–914.

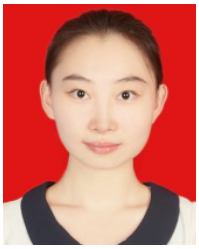
[29] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, ''The stanford CoreNLP natural language processing toolkit,'' in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguistics: Syst. Demonstrations*, 2014, pp. 55–60.

[30] W. Wu, H. Li, H. Wang, and K. Q. Zhu, ''Probase: A probabilistic taxonomy for text understanding,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, 2012, pp. 481–492.

YONGJUN LI (Member, IEEE) received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, China, in June 2007. From June 2007 to June 2009, he was a Postdoctoral Researcher with Peking University, China. From June 2012 to June 2013, he was a Visiting Scholar with the University of Massachusetts Amherst, MA, USA. He currently works as an Associate Professor with the School of Computer, Northwestern Polytechnical University. He has published over 30 papers as the main author, and most of them have been published in reputed journals and top international conferences. His current research interests include social computing, data mining, and network security.

WENLI JI received the B.Sc. degree in computer science and technology from the Xi'an University of Architecture and Technology, China, in 1997, and the M.S. degree in computer science and technology from the Xi'an University of Science and Technology, China, in 2004. She is currently an Associate Professor with the School of Communication and Information Engineering, Xi'an University of Science and Technology. Her current research interests include data mining and information security.

YAO DENG received the B.Sc. degree in communication and information engineering from the Xi'an University of Science and Technology, China, in 2016, where he is currently pursuing the M.S. degree in computer science and technology. His research interests include social computing and social networks.



XING GAO received the B.Sc. degree in communication and information engineering from the Xi'an University of Science and Technology, China, in 2016, where she is currently pursuing the M.S. degree in computer science and technology. Her research interest includes social computing.