A Location Inferring Model Based on Tweets and Bilateral Follow Friends

Xia Wang
College of Computer, Hangzhou Dianzi University, Hangzhou, China
Email: xiawang0707@gmail.com

Ming Xu*, Yizhi Ren, Jian Xu, Haiping Zhang, Ning Zheng
College of Computer, Hangzhou Dianzi University, Hangzhou, China
Email: {mxu, renyz, jian.xu, zhanghp, nzheng@hdu.edu.cn}

***Abstract*—Inferring user's location has emerged to be a critical and interesting issue in social media field. It is a challenging problem due to the sparse geo-enabled features in social media, for example, only less than 1% of tweets are geo-tagged. This paper proposes a location inferring model for microblog users who have not geo-tagged based on their tweets content and bilateral follow friends. An approach for extracting local words from "textual" data in microblog and weighting them is used to solve the sparse geo-enabled problem, and the maximum weight location vocabulary from his/her friends or tweets is inferred as the user's location. On a Sina-Weibo test set of 10,000 users from 10 cities, with 10 selected local words for each city, the inferring location accuracy on the city-level can reach 78.85%, and on the province-level can reach 81.39%. Compared with the TEDAS method, our method can achieve better accuracy.**

***Index Terms*—location inference, Sina-Weibo, local words**

I. INTRODUCTION

In recent years, microblog and microblog services, such as Twitter [1] and Sina-Weibo [2], Sina-Weibo the Chinese equivalent of Twitter [1], have seen a rapid growth. The term "microblog" means social platforms share the same features with Twitter. Users generate large quantities of data on the microblog platform which about what a person is thinking and doing in a particular location to share with his/her friends. The data are the messages that users can send and read text-based posts composed of up to 140 characters, called tweets [2].

This has spurred numerous research efforts to mine this data for various applications, such as event detection [3, 4, 5, 6, 7, 8, 9, 20] and news recommendation [11, 12]. Many such applications could benefit from information about the location of users, but unfortunately location information is currently very sparse. Only less than 1% of tweets are geo-tagged.

Two major challenges in the user location inference have yet to be fully addressed. First, because only less than 1% of tweets are geo-tagged the tweets' locations are very sparse. Second, the accuracy of the user location inferring problem is not entirely satisfactory.

In this paper, a method to infer the home, or primary, locations of microblog users from the content of their tweets and their bilateral follow friends is proposed to overcome this location sparseness problem, in the meantime, improving the accuracy. The goal is to infer location at the province-level and city-level. The benefit of the proposed method is two-fold. On the one hand, the output can be used to present information, recommend businesses and services and place-based advertisements that are relevant at a local level; On the other hand, our examinations of the discriminative features used by our algorithms suggest strategies for users to employ if they wish to microblog publically but not inadvertently reveal their locations.

Contributions in this work are as follows:

- An extracting and weighting local words approach based on microblog content is proposed to solve the location sparseness problem. The proposed approach can extract and weight geographic words efficiently from microblog content without the geo-tags.
- A locations inference model for microblog users based on information of tweets, bilateral follow friends and external location knowledge (e.g., dictionary containing names of cities and states) is proposed. This model not only can infer user location based on user tweets and bilateral follow friends, but also can infer only based on user tweets or bilateral follow friends.
- A comparative experiment with the TEDAS method indicates that our method can achieve better accuracy. On a Sina-Weibo test set of 10,000 users from 10 cities, with 10 selected local words for each city, the inferring location accuracy on the city-level can reach 78.85%, and on the province-level can reach 81.39%.

The rest of this paper is organized as follows: related work is in Section 2; Section 3 formalizes the problem of predicting a Twitter user's geo-location and describes several definitions; in Section 4, user's location inferring model is introduced; we present the experimental results

[1] www.twitter.com
[2] www.weibo.com

doi:10.4304/jcp.9.2.315-321

in Section 5; finally, conclusions and future work are discussed in Section 6.

II. Related Work

The location inference is an active field of research, and many unique methods have been proposed, including studies of blogs [13], web-pages [14, 15], and microblog [16, 17, 18, 19]. There are numerous research efforts to mine user location for various applications, such as event detection and news recommendation. Many such applications could benefit from information about the location of users, but unfortunately location information is currently very sparse. Only less than 1% of tweets are geo-tagged. Hence in this paper we focus on the microblog, especially on using "textual" data and bilateral follow friends, meaning no need for user IP information, or private login information.

Hecht et al. [16] attempted state-level location estimation using a Multinomial Naïve Bayes model to classify user location. Hecht et al. found that users implicitly reveal location information in their tweets, with or without realizing it. Our approach tried to extract location information from tweets on the province-level and city-level.

Cheng et al. [17] proposed a probabilistic framework for estimating a twitter user's city-level location based on tweet content. Their method correctly placed 51% of twitter users within 100 miles of their correct location. Chang et al. [18] used three probability models for locations, and compared both the Gaussian Mixture model (GMM) and the Maximum Likelihood Estimation (MLE). In their experiment, for 5,113 twitter users in the test set, with 250 selected local words or less was able to predict their home locations (within 100 miles) with the accuracy of 0.499.

Kinsella et al. [19] attempted to predict the location of an individual message by building language models of locations using coordinates extracted from geo-tagged twitter data. However our purpose is inferring the home location or primary location of the users, and we do not use geo-tags for the location estimation. The current knowledge of we know the best existing algorithms, directly related to ours, is created by Li et al. that used the location from the user's tweets and friends. Li et al. [20] predicted a user's location as the location from his/her friends or tweets that minimizes the overall distances between locations in his tweets and from his/her friends. They used this method in their event detection and analysis system named TEDAS. Our method not only uses user friends' location information but also uses user tweets to extract local words to overcome sparse challenge. Contrast to TEDAS, it only uses user friends' location information. Contrasted to other methods, our accuracy is better. In the meantime all the methods mentioned above are aiming at research on Twitter, all corpora are English. Comparing to our method which is focus on researching is based on Chinese corpora.

III. Extracting And Weighting Local Words Approaches

In this section, several definitions are defined in order to make an explicit distinction between the words which represents different location concepts.

Definition 1 Gazetteer words set *G*: G={ *g | g is a gazetteer word*}. A gazetteer word is the word belongs to a geographical gazetteer. In this work, it is limited as the geographical gazetteer of China [3]. Province-level gazetteer words contain provinces, municipalities, autonomous regions, Hong Kong and Macao. City-level gazetteer words contain prefectural-level city, municipalities, Hong Kong and Macao.

Definition 2 Local words set L_g: L_g={ *l | l is a local word, and l is a representative word for the place g*}. In this work, the top high weight words produced by a word extracting algorithm in Section 3 will be regarded as a local words set L_g, e.g. Hangzhou's local words set is { West Lake, Xiaoshan, Hangzhou, Yuhang, Zhejiang, Qianjiang, Alibaba, Hangcheng, cotton dress, down coat }. Let $L = \cup L_g$ ={ *l | l is a local word* }.

Definition 3 Geographic words set *Geo*: $Geo = G \cup L$ ={ *geo | geo is a geographic word* }. *Geo* is a geographic word set which contains gazetteer and local words. A place *g*'s geographic words is $Geo_g = \{g\} \cup L_g$ ={ *geo | geo is a geographic word, and geo is a representative word of the place g*}.

Definition 4 two-tuples set *X*: X ={<n_i, num_i>| n_i *is a nouns or local word, and* num_i *is the occurrence number of* n_i *in the text*}. Symbolizing $r_X(n_i) = \begin{cases} num_i & if < n_i, num_i > \in X \\ 0 & other \end{cases}$, $\sum X = \sum_{i=1}^{|X|} num_i$ means get the all nouns number, and $|X|$ means get the noun number in X.

Definition 5 Corpus *C*: C={ c_i | c_i *is the tweets messages from user i*}. The tweets messages from user *i* can be regarded as $c_i = < u_i, g_i, F_i, T_i >$, here u_i is user *i*, g_i is a geographic word in location profile of user *i* (user's location profile could be empty), two-tuples set F_i={<n_i, num_i >| n_i *is a noun form location profile of user i's bilateral friends, and numi is the* n_i*'s occurrence number in location profile of i's bilateral friends*}, two-tuples set T_i ={<n_i,num_i>| n_i *is a noun, and numi is the* n_i*'s occurrence number in tweet text from user i's tweet text*}.

The purpose of this work is to infer the location for those whose location profiles are empty.

A TFDFIDF algorithm which improved the original TFIDF [22] algorithm is proposed to extract and weight the local words. The TFDFIDF algorithm is described as the pseudo-code in Algorithm 1, and the notations use throughout the algorithm in Table I.

The TFDFIDF algorithm runs as following steps.

First, normalized frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the location (line 3). $N_g(n)$ denotes noun *n*'s appearing number when location is *g*. N_g denotes all nouns number when location

[3] http://www.china.com.cn/ch-quhua/

is g. Dividing the N_g by the $N_g(n)$ gives the normalized term frequency $tf_{g,n}$.

Second, document frequency based on the same locations (line 4). Take this step in order to make wider using terms have higher weight in the same location. $U_g(n)$ the user number with whose tweets contains term n when location is g. $|U_g|$ the user number in location g. Dividing the $|U_g|$ by the $U_g(n)$ gives the normalized term frequency $df_{g,n}$.

Third, inverse document frequency based on different locations (line 5). Take this step in order to make wider using terms have lower weight in the different locations. $|G|$ means location number, $NL_g(n)$ means the locations number which contains term n. Dividing the $NL_g(n)$ by the $|G|$ gives the normalized term frequency $idf_{g,n}$. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to 1+ $NL_g(n)$.

Finally, calculating the term n's *TFDFIDF* weight according to the $tf_{g,n}$, $df_{g,n}$ and $idf_{g,n}$ (lines 6-8). Put the term n and its weight in the $TFDFIDF_g$ (line 7). When finish traversing all locations returning *TFDFIDF* (line 11).

Then geographic words which contain local words and gazetteer words weighted as follow:

$$weight_g(n)=\begin{cases}1 & n\in(G-L)\\ \log(tfdfidf_{g,n}\cdot\delta) & n\in L_g\\ 0 & other\end{cases} \quad (1)$$

When term n belong to L_g the term weight is $\log(tfdfidf_{g,n}\cdot\delta)$, when term n belong to *G-L* the term weight is 1, else the term weight is 0. Here δ is a parameter, and it is set empirically. The detail will discuss in the later section.

In the Section 5 we use the TFDFIDF algorithm extracting the local words and their weight, and choose the top 10 local words in each province or city.

TABLE I.
NOTATIONS IN ALGORITHM 1

Notation	Meaning
$N_g(n)$	$N_g(n)=\sum_{i=1}^{\|C_g\|} r_{T_i}(n)$, $C_g=\{c_i \mid c_i=<u_i, g_i, F_i, T_i>, c_i\in C, g_i=g\}$. $N_g(n)$ denotes noun n's appearing number when location is g.
N_g	$N_g=\sum_{i=1}^{\|N\|} N_g(n)$, here N is set of all of nouns in text of tweets corpus C. N_g denotes all nouns in users' tweets when location is g.
$U_g(n)$	$U_g(n)=\{u_i \mid \exists c_i=<u_i, g_i, F_i, T_i>, c_i\in C, r_{T_i}(n)\neq 0, g_i=g,\}$, $U_g(n)$ is the user number with whose tweets contains term n when location profile is g.
U_g	$U_g=\{u_i \mid \exists c_i=<u_i, g_i, F_i, T_i>, c_i\in C, g_i=g,\}$, $\|U_g\|$ is the user number in location g.
$NL_g(n)$	$NL_g(n)=\sum_{g=1}^{\|G\|} ContainN_g(n)$, $ContainN_g(n)=\begin{cases}1 & n\in N_g\\ 0 & n\notin N_g\end{cases}$, $NL_g(n)$ is the locations number which contains term n

ALGORITHM I
TFDFIDF ALGORITHM

Algorithm 1: TFDFIDF Algorithm

Input: user, user tweets and user's location($C=\{c_i \mid c_i$ *is a user i's tweets message*$\}$)

Output: Local words and their weight (*TFDFIDF*)

/*Initialization $C=\{c_i \mid c_i$ *is a user i tweets message*$\}$ */

01 **for** each $g\in G$ do
02 **for** each $n\in N$ do
// N is set of all of nouns in text of tweets corpus C
03 $tf_{g,n}=N_g(n)/N_g$
04 $df_{g,n}=|U_g(n)|/|U_g|$
05 $idf_{g,n}=\log[|G|/(1+NL_g(n))]$
06 $tfdfidf_{g,n}=tf_{g,n}\cdot df_{g,n}\cdot idf_{g,n}$
07 $TFDFIDF_g=\cup\{tfdfidf_{g,n}\}$
// $TFDFIDF_g=\{tfdfidf_{g,n} \mid n\in N\}$
08 $TFDFIDF=\cup TFDFIDF_g$
// $TFDFIDF=\{TFDFIDF_g \mid g\in G\}$
09 **endfor**
10 **endfor**
11 **return** *TFDFIDF*

IV. LOCATION INFERENCE MODEL

In this section, a location inferring model based on the content of the user's tweets and bilateral follow friends depicted with the technical details. Bilateral Follow Friends which is a type of relationships on Sina-Weibo means that you are mutual friends [21]. Since a bilateral follow friend is your follower and friend at the same time, the set of bilateral follow friends is the intersection of the set of friends and the set of followers. The reason why choose bilateral follow friends to infer the users' location is that we find out a user contracts more tight with bilateral follow friends than other types of relationships.

To connect a user's location with locations from his/her tweets and bilateral follow friends, according to the following three observations. First, a user's location is more likely to appear in his/her tweets than other locations. Second, a user's bilateral follow friends tend to be closer with the user geographically. Third, a user's location is mentioned at least once in his/her tweets or is the same with at least one of his/her bilateral follow friends. With these observations, we can infer a user's location as the location from his/her friends or tweets that maximizes the weighted location.

Figure 1 shows the users' location inferring process. Since user i input the model, there is two-step strategy to get the final result. The two ways to inferring users' location mutual independence before the weight process module with the dotted box. One way needs obtain the user's bilateral follow friends' profile geographic words two-tuples set F_i. The other way is based on user's tweets, also using the method in Section 3 to extracting tweets geographic words two-tuples set T_i. In the weight process module, using bilateral follow friends' profile geographic words two-tuples set F_i and tweets geographic words two-tuples set T_i calculate the location weight according

to the formula 1 in Section 3. Finally, make the maximum weight location as the user's inferring location.

We propose the model finding the maximum weighted location according to the location inferring algorithm as pseudo-codes show in Algorithm 2.

The location inferring algorithm input the user *i*'s tweets and bilateral follow friends output the inferring location, runs as following steps.

First, traverse all the location to initialize the location *g*'s weight. Then traverse all geographic words in Geo_g get the geographic word *geo*'s weight $weight_g(geo)$ according to the formula 1 in last section (lines 3-8).

Second, get the geographic words appearing numbers then store the geographic words and their appearing numbers in T_i and F_i (lines 10-11). The appearing number *tcount* of term *geo* in user tweets in two-tuples set T_i (line 10). The appearing number *fcount* of term *geo* in user *i*'s bilateral follow friends' information in two-tuples set F_i (line 11). According to *geo* is representative to the place *g* calculate the geographic words' weight $w_{g,geo}$ then get the location weight w_g (lines 12-13). λ is a parameter , set empirically.

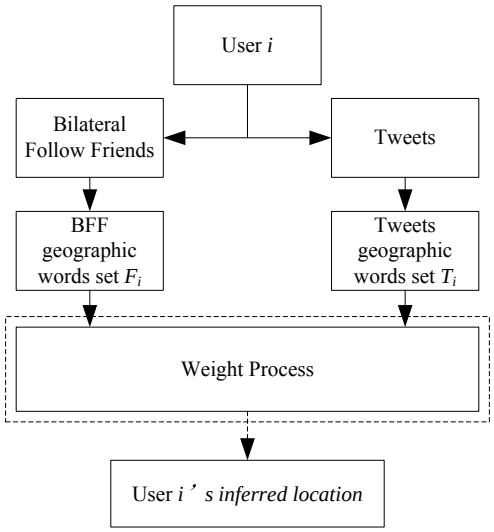

Figure 1. User's Location Inferring Process

Finally, find the geographic word *g* when the location weight w_g attains the largest value (line 17). Meanwhile, save the location in *MAXLocation*.

This location inferring model superiors to existing models because it relies on both user bilateral follow friends and tweets content. For instance, user *i* just moved from A city to B city. For this reason most of his/her microblog bilateral follow friends are in A city. But with his recent tweets mention about B city's traffic, weather, and sports team et al, this model can locate user *i* in city B.

ALGORITHM II
LOCATION INFERRING ALOGRITHM

Algorithm 2: Location Inferring Alogrithm

Input: user *i*'s tweets and bilateral follow friends' profile location information($c_i =< u_i, g_i, F_i, T_i >$)

Output: user *i*'s inferring location(*MAXLocation*)

/*Initialization *MAX*← ∅ , *MAXLocation* ← ∅ ,w_g← ∅ , *tcount*← ∅ ,*fcount*← ∅ */
//*MAXLocation is variable of the max weighted location;*
01 **for** $g \in G$ do // traversing all the location areas
02 $w_g \leftarrow \varnothing$; // initialize location *g*'s weight
03 **for** $geo \in Geo_g$ do
04 **if** $geo \in (G - L)$
05 **then** $weight_g(geo)$=1
06 **elseif** $geo \in L_g$
07 **then** $weight_g(geo) = \log(tfdfidf_{geo,n} \cdot \delta)$
08 **else** $weight_g(geo) = 0$
//when *geo* represents location *g* according to the formula 1
09 $w_{g,geo} \leftarrow \varnothing$;
// initialize *geo*'s weight when *geo* represents location *g*
10 tcount← $r_{F_i}(geo)$ // *tcount* is the appearing number of term *g* in F_i
11 *fcount*← $r_{T_i}(geo)$ //*fcount* is the appearing number of term *g* in T_i
12 $w_{g,geo} = \lambda * weight_g(geo) * tcount + weight_g(geo) * fcount$
// λ is a parameter, and is set empirically.
13 $w_g = w_g + w_{g,geo}$
14 **endfor**
15 *tcount*← ∅ ,*fcount*← ∅
16 **endfor**
17 $MAXLocation \leftarrow \arg\max_g(\{w_g \mid g \in G\})$
18 **return** *MAXLocation*

V. EXPERIMENTATION

In this section, an experimental study of location inference with local words which is extracting and weighting based on TFDFIDF algorithm is detailed. The goal of the experiments is to evaluate the local words extracting and weighting method based on TFDFIDF, the location inference method described in Section 4, and how the local words weight impacts the quality of inference.

Data for the experiments was originally collected between Dec. 2012 and Feb. 2013 by using Sina-Weibo's status streams and friendships streams APIs. Limit for the data is getting the recent 200 tweets if the user tweets more than 200, and getting at most 1,000 bilateral follow friends if the user bilateral follow friends are too many. The filter was used to retrieve all users whose followers were more than 100, tweets were more than 100, and profile location was from 10 prescribed cities which with high Sina-Weibo usage in the China continental. The cities were: Beijing, Shanghai, Tianjin, Chongqing, Hangzhou, Shenzhen, Guangzhou, Chengdu, Wuhan, and Shenyang. Meanwhile, the location checker repeated check the users' profile location to insurance the users' location accuracy during the experiment. The data set is randomly split into training (90%) and testing (10%) sets. The training set consists of 90,000 users with 17,768,919 tweets, and the testing set consists of 10,000 users with 1,973,141 tweets. The users are average from the ten cities which are mentioned above.

A. Evaluating the Local Words Extracting and Weighting

Preprocess the training set, 90,000 users with 17,768,919 tweets, with segmentation and tag the nouns

in the tweets. Then extract local words from the training set with the TFDFIDF algorithm described in Section 3. Empirically, this experiment chooses the top 10 maximum TFDFIDF value. Skip the place name's same level word is the local words chosen principle. Because the same level words will be an independent location. For instance, city Shenyang original local words contains same level city Dalian, Fushun, and Anshan, therefore, replace of three words whose TFDFIDF value less than these words.

Table II shows the top 10 city-level local words. Meanwhile, Table III shows that the top 10 province-level local words, city Guangzhou and Shenzhen comes from the same province, shows in the Table III there are 9 province and 90 local words. Noticed that province Liaoning chooses the three city-level words Dalian, Fushun, and Anshan because they are not on the same location level. As shows in the tables local words consist of several types. Geographical names are the largest component of local words, such as Hubei, Hankou, Wuhan, Wuchang e.g. in Wuhan's city level local words. Place of interests in the local words for example West Lake in the city Hangzhou, Qixia in the city Nanjing e.g. are both famous tourist resorts. Cultural features in the local words like crosstalk in municipality Tianjin, Cantonese in Guangzhou. In addition, famous persons' name, dialect words, sports teams and local news media are likely to appear in the local words.

TABLE II
CITY-LEVEL LOCAL WORDS(TOP 10)

City-level Local words(top10)	
Wuhan (武汉)	Hubei, Hankou, Wuhan,Wuchang, Jianghan, Huazhong, Wuhan University, Changjiang, East Lake, train station (湖北, 汉口, 武汉, 武昌, 江汉, 华中, 武大, 长江, 东湖, 火车站)
Tianjin (天津)	Binghai, Tianjin, Bohai, crosstalk, Taida, Nankai, Tanggu, Jinmen, Jincheng, Jianbing (滨海, 天津, 渤海, 相声, 泰达, 南开, 塘沽, 津门, 津城, 煎饼)
Shenzhen (深圳)	Shenzhen, Huaqiaocheng, Baoan, Nanshan, Huawei, Futian, Luohu, Huaqiang, estate, Coastal City (深圳, 华侨城, 宝安, 南山, 华为, 福田, 罗湖, 华强, 不动产, 海岸城)
Shenyang (沈阳)	Shenyang, Liaoning, Tiexi, Haolun, Zhao Benshan, Chinese business morning view, Yoshinoya, Style weekly, Early morning news, Taiyuan street (沈阳, 辽宁, 铁西, 皓伦, 赵本山, 华商晨报, 吉野, 时尚生活导报, 新闻早早报, 太原街)
Shanghai (上海)	Shanghai, Pudong, Fudan, Zhangjiang, Litte darling, Hongqiao, Hongkou, partner, Dream choir, Salon (上海, 浦东, 复旦, 张江, 囡囡, 虹桥, 虹口, 伙伴, 梦想合唱团, 沙龙)
Nanjing (南京)	Nanjing, Jinling, Suning, Xinjiekou, Jiangsu, Shuntian, Qixia, business hall, Modern Express, Xicihutong (南京, 金陵, 苏宁, 新街口, 江苏, 舜天, 栖霞, 营业厅, 现代快报, 西祠胡同)
Hangzhou (杭州)	West Lake, Xiaoshan, Hangzhou, Yuhang, Zhejiang, Qianjiang, Alibaba, Hangcheng, cotton dress, down coat (西湖, 萧山, 杭州, 余杭, 浙江, 钱江, 阿里巴巴, 杭城, 棉衣, 羽绒)
Guangzhou (广州)	Guangzhou, Huanan, Huagong, Panyu, Cantonese, Youth League Committee, senior fellow apprentice, senior sister apprentice, student union (广州, 华南, 华工, 番禺, 粤语, 团委, 师兄, 师姐, 广东, 学生会)
Chongqing (重庆)	Chongqing, Nanping, Jiefangbei, Dragon Lake, tea house, Shapingba, light rail, toddlers, young girl, Guanyingqiao (重庆, 南坪, 解放碑, 龙湖, 茶馆, 沙坪坝, 轻轨, 娃儿, 小妹, 观音桥)
Beijing (北京)	Guoan, Beijing, Peking University, Mao Zedong, capital, Chaoyangqu, Tsinghua, books, Yuan Yulai (国安, 北京, 北大, 毛泽东, 首都, 京城, 朝阳区, 清华, 图书, 袁裕来)

B. Estimating the Location Inference Model

This experiment is carried on the testing set, which consists of 10,000 users with 1,973,141 tweets. The users are average from the ten cities which are mentioned above. In this subsection, infer users' location with the location inference model described in the Section 4. Meanwhile compare this method with the baseline place name, Gazetteer method, and the TEDAS method [20]. Place name has been introduced in the Section 3. Gazetteer method means use gazetteer to infer the location. In the city level, use the smaller administrative region such as Futian, Luohu and Nanshan e.g. in city Shenzhen according to the geographic gazetteer of China[4].

TABLE III
PROVINCE-LEVEL LOCAL WORDS(TOP 10)

Province-level Local words(top10)	
Hubei (湖北)	Hubei, Hankou, Wuhan, Wuchang, Jianghan, Central China, Wuhan University, Changjiang, East Lake, train station (湖北, 汉口, 武汉, 武昌, 江汉, 华中, 武大, 长江, 东湖, 火车站)
Tianjin (天津)	Binghai, Tianjin, Bohai, crosstalk, Taida, Nankai, Tanggu, Jinmen, Jincheng, Jianbing (滨海, 天津, 渤海, 相声, 泰达, 南开, 塘沽, 津门, 津城, 煎饼)
Guangdong (广东)	Shenzhen, Huaqiaocheng, Huawei, estate, Coastal City, Guangzhou, Huanan, Cantonese, Guangdong, student union (深圳, 华侨城, 华为, 不动产, 海岸城, 广州, 华南, 粤语, 广东, 学生会)
Liaoning (辽宁)	Shenyang, Liaoning, Tiexi, Haolun, Chinese business morning view, Yoshinoya, Taiyuan street, Dalian, Fushun, Anshan (沈阳, 辽宁, 铁西, 皓伦, 华商晨报, 吉野, 太原街, 大连, 抚顺, 鞍山)
Shanghai (上海)	Shanghai, Pudong, Fudan, Zhangjiang, Litte darling, Hongqiao, Hongkou, partner, Dream choir, Salon (上海, 浦东, 复旦, 张江, 囡囡, 虹桥, 虹口, 伙伴, 梦想合唱团, 沙龙)
Jiangsu (江苏)	Nanjing, Jinling, Suning, Xinjiekou, Jiangsu, Shuntian, Modern Express, Yangzhou, Suzhou, Wuxi (南京, 金陵, 苏宁, 新街口, 江苏, 舜天, 现代快报, 扬州, 苏州, 无锡)
Zhejiang (浙江)	West Lake, Xiaoshan, Hangzhou, Yuhang, Zhejiang, Qianjiang, Alibaba, Hangcheng, Ningbo, Wenzhou (西湖, 萧山, 杭州, 余杭, 浙江, 钱江, 阿里巴巴, 杭城, 宁波, 温州)
Chongqing (重庆)	Chongqing, Nanping, Jiefangbei, Dragon Lake, tea house, Shapingba, light rail, toddlers, young girl, Guanyingqiao (重庆, 南坪, 解放碑, 龙湖, 茶馆, 沙坪坝, 轻轨, 娃儿, 小妹, 观音桥)
Bejing (北京)	Guoan, Beijing, Peking University, Mao Zedong, capital, Chaoyangqu, Tsinghua, books, Yuan Yulai (国安, 北京, 北大, 毛泽东, 首都, 京城, 朝阳区, 清华, 图书, 袁裕来)

[4] http://www.china.com.cn/ch-quhua/

In the city level experiment empirical set the parameter $\sigma = 1.31*10^{17}$ in order to ensure the local words average weight equals to 12. The details will explain in the next subsection. The City-level relationship between parameter λ and precision is shown in the Figure 2. Inferring users' location only rely on tweets when λ=0 and inferring only rely on bilateral follow friends' information when λ=1. In the Figure 2, the precision is changing with the λ, and when λ equals 0.8 it get the peak. Obviously, our method is better than baseline place name, Gazetteer, and the TEDAS method, especially, when the λ=0.8 the precision is 78.85%.

In the province level experiment empirical set the parameter $\sigma = 3.87*10^{8}$ in order to ensure the local words average weight equals to 3.5. The Province-level relationship between parameter λ and precision is shown in the Figure 3. The same as city level when λ equals 0.8 the precision get the peak, baseline place name get 79.07%, Gazetteer get 81.32%, and TFDFIDF method get 81.39%. These three methods are all better than the TEDAS method which precision maximum is 69.12%.

In this subsection we learn that the precision of Gazetteer and TFDFIDF method are very similar especially in the province level. The reason why is that their local words are very similar, many Gazetteer words are on the local words list, maybe not in the top 10, but also at very top of the list.

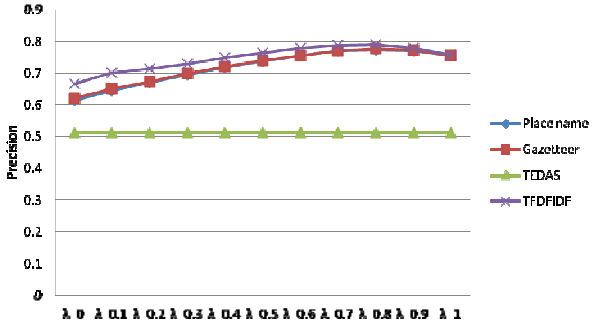

Figure 2. City-level relationship between parameter λ and precision

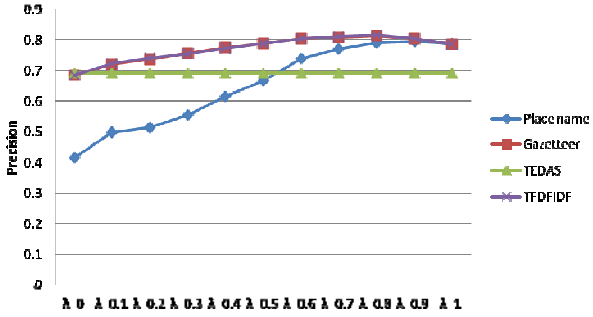

Figure 3. Province-level relationship between parameter λ and precision

C. Estimation Local Words Weight

An important question remains: how did the local words weight influence the quality of inference. In the city level experiment we set the average weight 12.0, and 3.5 in the province level experiment. To illustrate the impact of an increasing value of local words weight, we begin with a specific experiment using the testing dataset.

The city-level and province-level relationship between words weight and precision is illustrated in the Figure 4. According to the chart the precision increased rapidly when the local words average weight was less than 2.0, then, remained stable when the weight was heavier than 1.5 both in city-level and province-level. With this result we can come to a conclusion that the local words weight has less influence on the locate precision when the weight is greater than 2.0.

VI. Conclusions And Future Work

In this paper, an algorithm named TFDFIDF algorithm is presented to extracting local words and their weight to overcome the location sparseness problem efficiently. Furthermore, we proposed a location inferring model based on the content of the Sina-Weibo user's tweets and user bilateral follow friends. Experimental performance demonstrates that our model achieves higher performance than the current knowledge of we know the best existing algorithms for inferring locations of microblog users both based on user tweet content and bilateral follow friends information. We take our experiment in city-level and province-level and explore the relationships between the local words weight and the inference precision.

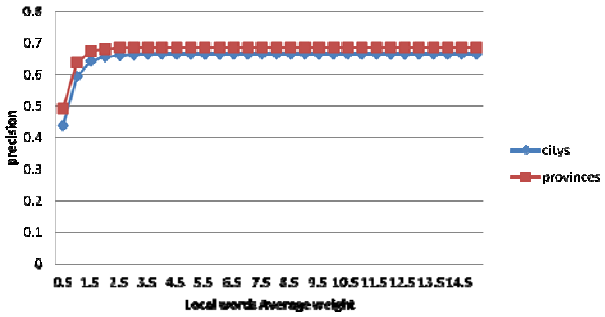

Figure 4. the relationship between Local words average weight and precision

Much future work has arisen from this study of users' location inference. With regard to the local words extracting and weighting method, we are looking into including heuristic method into our algorithm. We also are working to extend our inference experiments to smaller granularities such as neighborhood level. Considering the accuracy of users' profile location, we are thinking about taking LBS information into our model.

Acknowledgment

This work is supported by the Natural Science Foundation Natural Science Foundation of China under Grant No.61070212, 61003195 and 61100194, the Zhejiang Province Natural Science Foundation Natural Science Foundation of China under Grant No.Y1090114 and LY12F02006, the Zhejiang Province key industrial projects in the priority themes of China under Grant No 2010C11050, the science and technology search planned projects of Zhejiang Province (No.2012C21040) and the Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Y201120356).

REFERENCES

[1] Wang, Dong, et al. "The pattern of information diffusion in microblog."Proceedings of The ACM CoNEXT Student Workshop. ACM, pp. 3, 2011.

[2] Guo, Zhengbiao, Zhitang Li, and Hao Tu. "Sina microblog: an information-driven online social network." Cyberworlds (CW), 2011 International Conference on. IEEE, pp. 160-167, 2011.

[3] Kumaran, Giridhar, and James Allan. "Text classification and named entities for new event detection." Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 297-304, July 2004.

[4] Lampos, Vasileios. "Detecting events and patterns in large-scale user generated textual streams with statistical learning methods." arXiv preprint arXiv:1208.2873, 2012.

[5] Lampos, Vasileios, and Nello Cristianini. "Nowcasting events from the social web with statistical learning." , 2011.

[6] Lee, Ryong, Shoko Wakamiya, and Kazutoshi Sumiya. "Discovery of unusual regional social activities using geo-tagged microblogs." World Wide Web 14.4, pp. 321-349, 2011.

[7] Watanabe, Kazufumi, et al. "Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs."Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, pp. 2541-2544, 2011.

[8] Wu, Dongqing, Fengjian Yang, and Chaolong Zhang. "Statistical Methods based on Semantic Similarity of Topics Related to Microblogging." Journal of Software 8.1 (2013): 192-199.

[9] Xin, Mingjun, Hanxiang Wu, and Zhihua Niu. "A Quick Emergency Response Model for Micro-blog Public Opinion Crisis Based on Text Sentiment Intensity." Journal of Software 7.6 (2012): 1413-1420.

[10] Wang, Xiaodong, and Juan Wang. "A Method of Hot Topic Detection in Blogs Using N-gram Model." Journal of Software 8.1 (2013): 184-191.

[11] Sankaranarayanan, Jagan, et al. "Twitterstand: news in tweets." Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 42-51, 2009.

[12] Bao, Jie, Mohamed F. Mokbel, and Chi-Yin Chow. "GeoFeed: A Location Aware News Feed System." Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, pp. 54-65, 2012.

[13] Fink, Clay, et al. "Geolocating blogs from their textual content." Working Notes of the AAAI Spring Symposium on Social Semantic Web: Where Web. Vol. 2. 2009.

[14] Amitay, Einat, et al. "Web-a-where: geotagging web content." Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 批 pp. 273-280, 2004.

[15] Buscaldi, Davide, and Paolo Rosso. "A comparison of methods for the automatic identification of locations in wikipedia." Proceedings of the 4th ACM workshop on Geographical information retrieval. ACM, pp. 89-92, 2007.

[16] Hecht, Brent, et al. "Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles." Proceedings of the 2011 annual conference on Human factors in computing systems. ACM, pp. 237-246, 2011.

[17] Cheng, Zhiyuan, James Caverlee, and Kyumin Lee. "You are where you tweet: a content-based approach to geo-locating twitter users." Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, pp. 759-768, 2010.

[18] Chang, Hau-wen, et al. "@ Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage." Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on. IEEE, pp. 111-118, 2012.

[19] Kinsella, Sheila, Vanessa Murdock, and Neil O'Hare. "I'm eating a sandwich in Glasgow: modeling locations with tweets." Proceedings of the 3rd international workshop on Search and mining user-generated contents. ACM, pp. 61-68, 2011.

[20] Li, Rui, et al. "TEDAS: a twitter-based event detection and analysis system."Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, pp. 1273-1276, 2012.

[21] Wang, Alex Hai. "Don't follow me: Spam detection in twitter." Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on. IEEE, pp. 1-10, 2010.

[22] Bun, Khoo Khyou, and Mitsuru Ishizuka. "Topic extraction from news archive using TF* PDF algorithm." Web Information Systems Engineering, 2002. WISE 2002. Proceedings of the Third International Conference on. IEEE, pp. 73-82, 2002.

Xia Wang received the B.S. degree in communication engineering from the Hangzhou Dianzi University in 2011. She is currently a master candidate in computer technology from the Hangzhou Dianzi University, P. R. China. Her research interest includes Intelligent Information Processing System, Social Network and Data Mining.

Ming Xu is a Professor in the college of Computer, Hangzhou Dianzi University, P. R. China. He received the doctor degree in computer science and technology from the Zhejiang University in 2004. His research interests include Digital Forensics, Network Security, Social Network and Data Mining.

Yizhi Ren is a Lecturer in the college of Computer, Hangzhou Dianzi University, P. R. China. He received the doctor degree in computer science and technology from the Dalian University of Technology in 2011. His research interests include Network Security, Social Computing and Evolutionary game.

Jian Xu is a Professor in the college of Computer, Hangzhou Dianzi University, P. R. China. He received the doctor degree in computer science and technology from the Zhejiang University in 2004. His research interests include Location-based Services, Mobile Computing, Distributed Database and Network Security.

Haiping Zhang is an Associate Professor in the college of Computer, Hangzhou Dianzi University, P. R. China. He received the master degree in Computer Software and Theory from the Hangzhou Dianzi University in 2005. His research interests include Digital Forensics, Network Security, Social Network and Corporate Information Technology.

Ning Zheng is a Professor in the college of Computer, Hangzhou Dianzi University, P. R. China. His research interests include Network Security, CAD, and CAM.