Improving Geocoding of a Twitter User Group using their Account Creation Times and Languages

Aleksey Panasyuk
Information Fusion Technology Branch
Air Force Research Lab
Rome, New York, USA
aleksey.panasyuk@us.af.mil

Kishan G. Mehrotra, Edmund Szu-Li Yu
Dept. of Electrical Engineering
and Computer Science
Syracuse University
Syracuse, New York, USA
mehrotra@syr.edu, esyu@syr.edu

***Abstract*—This paper proposes a classifier for predicting the location of followers of a Twitter influencer using their account creation times. Time is a universal feature and thus it can be used to characterize influencers from different parts of the world. In addition, the language of the location is used to improve classification performance. In the first step of the proposed two-step approach influencer's followers' account creation times are used to create a time distribution to predict the time zone's Coordinated Universal Time (UTC) offset. In the second step, language features are used to constrain locations associated with the UTC offset. The approach is illustrated by categorizing 320K Twitter celebrity influencers by their country. High confidence influencer predictions are used as training data for an improved geocoder. This geocoder automatically learns popular ways that Twitter users refer to locations within the country and can handle foreign alphabets.**

***Index Terms*—Location-aware influence, Geocoding using Time and Language, Account Creation Time Distribution**

I. Introduction

Twitter is an important broadcasting platform where a user's messages and profile information are made available to all others. Location information is important for understanding the content posted by hundreds of millions of Twitter users. High-level location data can benefit a study to analyze a specific geographic region.

Due to privacy reasons, and because all information on Twitter is public, home location is not typically disclosed. Previous research has proposed methods for improving geocoding from self-reported textual location, inferring it from user's friends/followers, or based on message traffic. Existing methods typically work with textual information collected using profile information or message traffic. Moreover, a universal geocoding solution that can handle various languages and their associated alphabets, popular Twitter slang, and purposely ambiguous phrases is inherently hard to build. As a result, most methods are illustrated on a small portion of the world with English as the primary language.

In this paper, a time-based method is proposed to characterize the location. The benefits of the time-based features are that they are not dependent on a specific alphabet or language and hence are universally applicable to the whole Twittersphere. We are not aware of any other research that attempts to strictly use time and language for this task. The methods in this paper can augment existing research that deals with influence; helping to characterize the followers since an influencer may have a large following, but be unknown in the geographic area of interest.

This research has many applications. For example, one application is for content recommendation by identifying geo-influencers that serve a specific time zone. Local traffic, local businesses, local news, and others are examples of geo-influencers. The geo-influencers and communities around them provide insight into the news and the reactions happening in the geographic region. Another application is where the influencer's self-reported location or topics of discussion targets a different area then the one inferred from followers. An influencer from a foreign country that is continuously engaged in attempting to influence local discussions associated with another country may be a potential disinformation campaign.

The rest of the paper is structured as follows. Section II presents related research. Section III presents the datasets. The time distribution and features that can be extracted from it are discussed in Section IV. The approach for predicting influencer's geographic region, using time and language information, is presented in Sections V and VI, respectively. Section VII shows the full pipeline for categorizing a large influencer set. Section VIII uses labeled influencers for training a location based geocoder followed by conclusions and future work.

II. Related Research

The problem explored is related to understanding geographic area of a user group and how time and language can complement it. This problem is related to location-aware influence maximization (LAIM) whereby a set of users is identified that has the biggest impact on the underlying geographic population [1].

There are three types of Twitter-related locations: user's home location, the location from where the tweet (message) is originating from, and location mentioned in a tweet [2]. For predicting a user's home location message traffic, profile information, or network structure may be explored.

Inkpen, et al. [13] develop a city, province, and country classifier for monitoring places mentioned in Twitter messages. The issue with message traffic is that only a small percent of it

contains location information. It may mention a location that is not indicative of the user's true home location, for example, a message posted when the user is traveling [3]. Some users are passive consumers of information and do not post a single message. A lot of information would be missed if passive consumers are not counted.

About a third of all users report their location via a textual string in their profile [4]. This textual string needs to be geocoded to latitude and longitude coordinates. Factors complicating geocoding include abbreviations, misspellings, blank textual field, and use of multiple alphabets from different languages. The location field may refer to something purposely ambiguous such as 'The Internet' or 'My House'. Attempts to fix these problems include: (i) correcting misspellings and abbreviations [5], (ii) focusing on users that employ high confidence locations consisting of known (city, province) pairs [6], and (iii) using a commercial geocoder [7]. But commercial geocoders make mistakes by making associations based on business name [8]. For users whose location cannot be extracted from message or profile information the connections that the user has made to others come into play. For a user whose self-reported location cannot be geocoded Compton, et al. [9] propose to use the median location of user's friends.

Existing methods are typically illustrated on English data [2]. To apply the methods to other languages, a universal geocoder is required but there does not exist a single solution for handling different languages.

Account creation time has been used in earlier work by some researchers; for example, to study network evolution by measuring how active the topics and users are. Reza, et al [11] explore the relationship between the Twitter creation time and the time of the last tweet and find that most users become less active content generators. Meeder et al. [12] illustrate that the dynamics between celebrity users and followers could be estimated using the celebrity's follower list and account creation times associated with each user. Account creation time, by itself, has not been utilized for predicting location [2].

Other features useful for understanding location are the time zone and UTC offset [14, 15]. Zannettou, et al. [16] used time zone to understand the targeted audience by tweets from Russian linked accounts. But due to privacy reasons Twitter has made these fields private in 2018. Time zone is a user-specified field that was available for only about a quarter of all users. Since Twitter no longer allows collecting time zone and UTC, and because other researchers have found these features useful in the past, therefore, there is a need for new features that can act in their place.

The research in this paper does just that by analyzing time distributions from account creation times of a user group and using those for estimating UTC offset as well as proposing other features that can help better understand influence.

III. Datasets

Two datasets were assembled for performing this research. The first dataset is made of user groups grouped by their self-reported textual location. The UTC offset associated with the time zone observed at the location serves as the label for the group. The second dataset, consisting of followers of 320K influencers, is used for illustrating the time and language classifier developed using the first dataset. High confidence predictions over second dataset are used for training an improved geocoder. More details are provided below.

A. UTC Offset Dataset

On Twitter, a user's profile may specify a self-reported location. In this dataset, consisting of over 377 million user profiles, user groups are binned by the self-reported location. This dataset consists of locations specified in English.

All self-reported locations are turned to lowercase with punctuation and spacing stripped out. Of particular interest are those self-reported locations that match (i) City + Province or (ii) City + Country Name from GeoNames. The city, country pairs are unique in that there are no other cities within the country with the same city name. Population of all cities considered in this dataset is over five thousand. Major well-known city names are included (without the country name) provided the city has a population of over 1 million. There are 357 such cities in this dataset. Each self-reported location was used by at least 250 unique users to ensure a large enough sample size for estimating the time distribution described in Section IV. This dataset consists of 12,271 user groups. Table I shows the five biggest user groups and their UTC offset label, denoted UTC^L, using equation (1).

TABLE I
Five biggest user groups in UTC Offset Dataset

Location	Group Size	UTC^L	Country
london	2065562	0.667	GBR
losangelesca	1768898	-7.333	USA
newyorkny	1425330	-4.333	USA
chicagoil	1173340	-5.333	USA
parisfrance	1026459	1.667	FRA

In equation (1), GeoNames is used to get the location's time zone via function tmz[1]. UTC and DST function gives the UTC offset observed at time zone during standard and daylight saving time, respectively (for time zones where daylight saving not observed UTC and DST produce equal offset). Daylight saving time is typically observed for eight months of the year and is thus given a larger weight.

$$UTC^L(loc) = \frac{1}{3}UTC(tmz(loc)) + \frac{2}{3}DST(tmz(loc)) \quad (1)$$

In our dataset, UTC^L takes 42 possible values ranging from -9.9 to 13.53. The corresponding UTC offset interval for our dataset is therefore [-10, 14) (UTC offset -12 and -11 exist, but belong to sparsely populated islands and therefore not part of the dataset).

[1]download.geonames.org/export/dump/timeZones.txt

B. *Influencers-Dataset*

In this dataset user groups are binned by the influencer they follow. The influencers for the dataset were chosen from the special Twitter account *@verified*. This account tracks all influencers that have passed an internal Twitter check (Twitter performs a special check on a verified celebrity and identifies them via a special blue badge). In this dataset, collected in the spring of 2019, there are 320,166 influencers. Due to Twitter API limits, for each influencer only a single API call was made which returned at most 5000 followers.

The ground truth consists of the country associated with the self-reported location reported by the influencer. It is checked whether this country can be used as a label, based on whether this country matches (i) the most frequent country from self-reported locations of followers and (ii) whether it is contained within the set of countries that would be predicted using followers' time and language features. It is shown that time and language features can be used to improve precision of the country labels. The country label and the associated influencer's followers' self-reported locations are used for training and illustrating an improved geocoder.

IV. Features from Time Distribution

The account creation time is always available, and is accurate, time-stamped by the system, and not self-reported by the user.

Given a group of users G, the elapsed time[2] measured in seconds, is used to generate a histogram with bin size s seconds. The histogram is converted into a frequency distribution, called normalized time distribution; where $f(t)$ represents the relative frequency of users that create their account within t^{th} interval.

About a third of the 24-hour cycle is expected to be devoted to sleep. During sleep time, the users in set G are less likely to have generated their accounts. The sleep portion of the 24-hour cycle can be identified by focusing on account creation times, which in turn can be identified if $f(t)$ is below the 33^{th} percentile in a normalized time distribution. We believe that if the users in G belong to a single location then the sleep time for the group (i) will be part of a single segment and (ii) will have the form of a U-shaped parabola. If a parabola cannot be fitted, it is indicative of a user group not being concentrated in the same time zone. The normalized time distribution may be uneven especially for a small group or small sample size, see blue lines in Fig. 1. To achieve smoothness and derive meaningful features we use n-points moving average of frequencies. In Fig. 1, frequency distributions obtained in this manner, using $n = 5$, are depicted by green lines and clearly they are much smoother.

In the rest of the paper $f(t)$ refers to the frequency obtained using the moving average over n observations.

First we identify whether, in set G, the sleep time occupies a single continuous segment which can be identified if there are only two intersection points per 24 hours (four per 48 hours) between time distribution curve and the line representing 33^{th} percentile[3]. The second step is to identify the sleep cycle. The start and end of the sleep cycle are the intersection points with a negative and positive slope, respectively. Our approach searches for the first two intersection points that meet this criteria over the 48 hours.

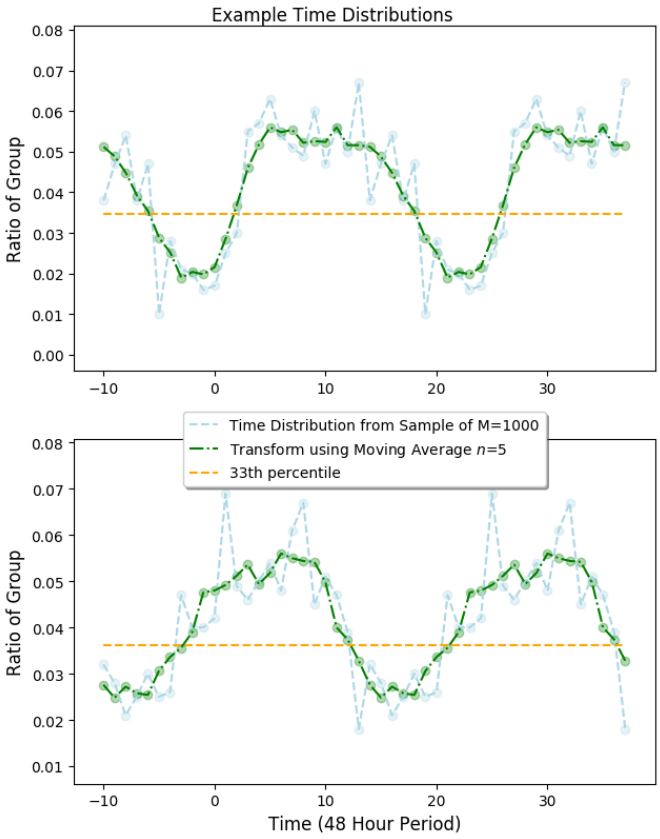

Fig. 1. Normalized time distributions for followers of Chicago Tribune (top chart) and BBC News (bottom chart) are shown. The blue line is the time distribution using $n = 1$ and the green line represents the distribution using the moving average with $n = 5$; moving average applied to smooth out irregularities. All frequency distributions are over 48 hours, as explained in the text. These distributions were obtained using a sample of size $M = 1000$. The potential sleep cycle is a continuous set of points that is below the orange line; obtained from the green curve.

Fig. 1 shows the time distributions of two user groups; followers of influencers *@ChicagoTribune* (top) and *@BBCNews* (bottom), respectively. The 24-hour time distribution is repeated over a 48-hour period where the values below the orange line represent a potential sleep cycle. Both distributions pass the first test in that there are four intersection points over a 48-hour period. The sleep cycle for *@ChicagoTibune* occurs between the first and second intersection points and for *@BBCNews* it is between the second and third intersection points. The third step is to fit a polynomial function and to test whether it is a U-shaped parabola. If it is a U-shaped parabola

[2]Twitter account creation time provides the time as well as the date. We consider only the time part of it, computing elapsed time in seconds from it.

[3]intersection point identified using python library NumPy with the command np.argwhere(np.diff(np.sign(f - g))).flatten()

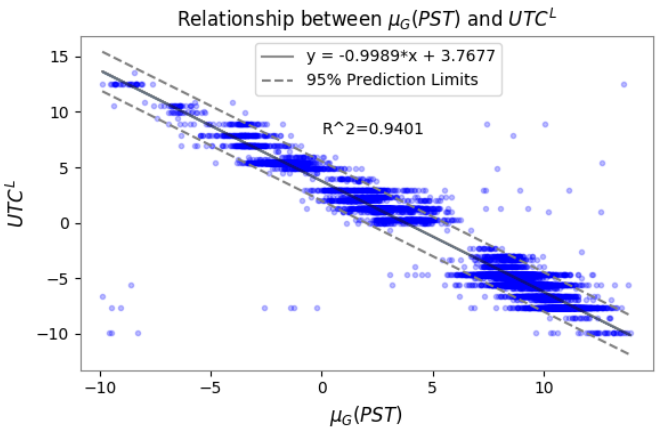

Fig. 2. Computed μ_G(PST) has a linear relationship with UTC^L indicating that the time distribution can be used for predicting the time zone a user group is from. Chart computed using $M = 250$, $p = 33$, $N = 100$, and $n = 5$.

the fourth step is to derive the minimum of the parabola defined as the Peak Sleep Time (PST). PST corresponds to a time when a user from the user group is least likely to have created their account. Due to concatenating two 24-hours cycles, PST can take a value greater than 24 (as seen for *@BBCNews*). If PST is greater than or equal to 24 we subtract 24 to reduce it to the range [0, 24).

Instead of using the entire $|G|$ observations of the group, we use a method akin to bootstrapping (see Efron and Tibshirani [19]). Random samples of size M are drawn from G, N times, and for each sample the PST is calculated. Finally, the average and standard deviation of PSTs over N samples is calculated; denoted by μ_G(PST) and σ_G(PST), respectively.

In calculating μ_G(PST) and σ_G(PST) the cyclical nature of the time is adjusted.[4] In our procedure, μ_G(PST) and σ_G(PST) are computed using [0, 24) interval. Finally, given a $\mu_G(\text{PST}) \geq 14$ the transformation $\mu_G(\text{PST}) - 14$ is applied to match the UTC^L interval [-10, 14).

In Fig. 2, points in the scatter plot represent group μ_G(PST) values. All groups from UTC offset dataset are represented. Each μ_G(PST) is computed using sample size $M = 250$, percentile $p = 33$, and moving average window $n = 5$ using $N = 100$ iterations. Figure shows that the μ_G(PST) has a strong linear relationship with UTC^L because observed $R^2 = 0.9401$.

For brevity, from now onward, we use PST to denote μ_G(PST). The next section examines the best parameters for computing PST that best aligns to the UTC^L associated with the location that a group of users belongs to.

V. Predicting UTC Offset

The goal of this section is to develop a procedure to predict the UTC_L of the location associated with user group G. For training the classifier the UTC offset dataset is utilized which consists of 12,271 user groups. PST, described above, is our primary feature in developing the prediction.

A. Parameter Determination

Choice of the sample size M, number of samples N, the size of moving average window n, and percentile p below which relative frequency $f(t)$ is considered to belong to sleep cycle are crucial in determining appropriate value of PST. In order to find the most appropriate values of these parameters we performed experiments for $M = [100, 250, 500, 1000]$, $N = 100$, $n = [1, 2, ..., 7, 8]$ and $p = [20, 25, 30, 33, 35, 40, 45]$. A straight line fit was performed for PST vs. UTC_L using the least squares estimation in the same way as in Fig. 2. To measure the performance of selected values of the parameters *Recall, Precision,* and *F1* measures were calculated; where,

$$\text{Recall} = \frac{\text{\# of user groups where PST-estimate calculated}}{\text{the number of user groups}},$$

$$\text{Precision} = \frac{\text{\# of correct UTC predictions}}{\text{\# of UTC predictions}},$$

and

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}.$$

Initially, predictions that were more than $t_1 = 0.5$ away from UTC^L were marked as incorrect. The μ_G(PST) and σ_G(PST), for all correct predictions were evaluated as 0.3318 and 0.0794, respectively. This provides a threshold $t_3 = 0.3318 + 2 \times 0.0794 = 0.4906$ beyond which the group standard deviation is considered too large. Such a group is considered too noisy to generate an accurate PST and, therefore, associated prediction is not considered valid. Out of 12,271 locations 1149 (9.36%) were identified as noisy. When we evaluated the PSTs for these noisy groups 65.36% produced a PST that is indeed more than $t_1 = 0.5$ away from its UTC^L.

Fig. 3 shows F1 for different values of p and n for $M = 250$ and $t_1 = 0.5$ over all user groups in the UTC offset dataset. It can be seen that the best performance is achieved using $p = 33$ and $n = 5$. The plot in Fig. 2 uses these best performing parameters.

Because we are interested in generating high confidence predictions so as to use those as training data for an improved geocoder the next two figures are focused on precision instead of F1. In Fig. 4 we plot the precision versus p for four different values of M, and in Fig. 5 we plot the precision versus n, respectively. In Fig. 5 we focus on groups of size 1000 or more (because 1000 is the biggest M value). From Fig. 4 it is clear that the precision is highest for $p = 33$, for all values of M. This value of p is also an intuitive choice because, as mentioned earlier, about a third of the 24-hour period is expected to be devoted to sleep. Smaller percentile ($p < 30$) reduces the associated sleeping cycle and it is harder to fit a parabola and to get a good UTC offset prediction. On the other hand, if p is too high (p≥40) then points that are outside of the sleeping cycle will be incorporated causing the performance to suffer.

[4]For example, average of PST values 0 and 23 = 11.5 is incorrect because in a 24-hours cycle [0,24), 23 is followed by 0; accurate average is 23.5.

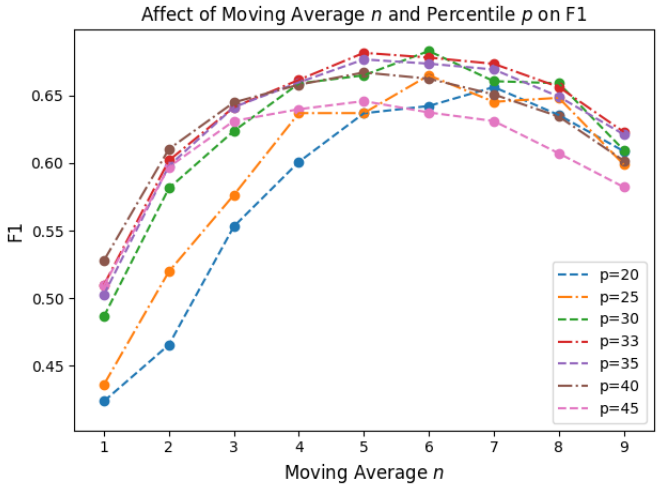

Fig. 3. Relationship between percentile p and n on F1 measuring overall ability to predict UTC^L ($t_1 = 0.5$, $M = 250$). The highest F1 = 68.14% is at $p = 33$ and $n = 5$ with Precision = 55.05% and Recall = 89.41%.

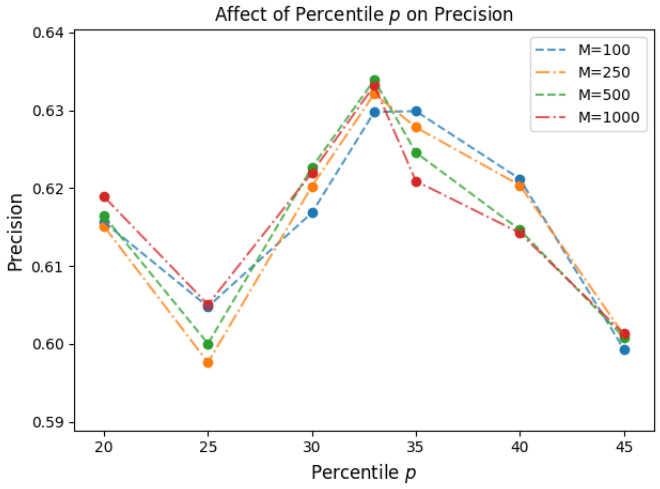

Fig. 4. Percentile $p = 33$ has the best overall precision across different sample sizes M (plot constructed using moving average $n = 5$ as it is associated with best F1 score).

The effect of n is to smooth out jumps in frequency distributions. From Fig. 5 it is clear that $n \in [2, 5]$ exhibit high precision for all values of M. From this figure we conclude that any choice of $n \in [2, 5]$ is reasonable to smooth out irregularities, preserve high precision, but is not too high to delete the sleep cycle from the time distribution. On the other hand, Fig. 3 shows that the F1 peaks at $n = 5$ and then starts to gradually decrease. Therefore, considering both, the precision and F1, we conclude that $n = 5$ is the best choice.

Similar performance is exhibited across different sample sizes, but $M = 500$ is the best performing for $p = 33$ and $n = 5$ in both figures and thus it is the variable of choice.

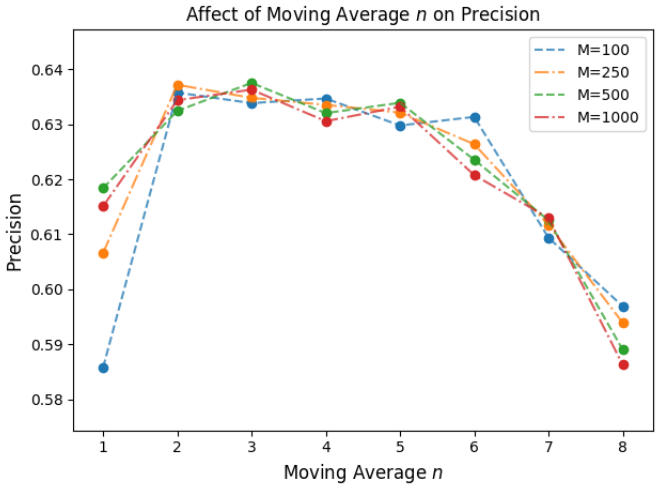

Fig. 5. Moving average $n = [2, 5]$ has the best overall precision across different sample sizes M (plot constructed using percentile $p = 33$ as it was associated with best F1 score).

B. The UTC Offset Identifier

If we are unable to find a U-shaped behavior in a normalized time distribution, then the associated influencer is identified as global, else an influencer is local. If $\sigma_G(\text{PST}) \geq t_3 = 0.49$, then the group G is considered noisy to make a reasonable PST estimate.

From previous subsection the best parameters for identifying a U-shaped behavior were $M = 500$, $N = 100$, $n = 5$, $p = 33$. The UTC offset is predicted using equation (2) which corresponds to line fitted using above choices of the parameters with $R^2 = 0.97717$. Table II shows the performance of this predictor for different thresholds t_1.

$$UTC^P = -1.0338 \times PST + 3.9988 \tag{2}$$

TABLE II
PRECISION OF UTC^P WITHIN t_1 OF UTC^L

Errors	Matches	Precision	t_1
5074	889	14.91	0.1
3815	2148	36.02	0.25
2183	3780	63.39	0.5
1204	4759	79.81	0.75
679	5284	88.61	1
376	5587	93.69	1.25
201	5762	96.63	1.5
89	5874	98.51	1.75
39	5924	99.35	2

VI. IMPROVEMENT VIA LANGUAGE

In this section we utilize language to further improve the performance of identifier proposed in the previous section. Given a time distribution associated with user group we can compute UTC^P. Let U_1 equal the set of countries whose cities have a time zone that observes UTC offset in range $[UTC^P - t_2, UTC^P + t_2]$, where t_2 is

a preselected threshold. For example, the set of countries [TLS, PLW, JPN, MNP, FSM, GUM, IDN, AUS, PRK, RUS, KOR, PNG], correspond to UTC offset range $\in$ [8, 10].

Our next step is to incorporate language information to constrain the set of possible countries for selected UTC offset range. Initially, the CIA World Factbook was utilized for this purpose. But for some countries the languages from CIA World Factbook do not align to the languages used on Twitter. For example, English is the most popular language in India for communication, but it does not make it into popular spoken languages (Hindi, Bengali, and others).

We utilized language preferences, a user-selected option, which is available for more than 99% of collected users on Twitter. In our dataset, there are 76 unique language codes representing 143 countries, each language was used by at least 100 users, collected over 377 million user profiles.

Given a user group, the users' language preferences are used to generate a language distribution, i.e., we calculate

$$g_G(\ell) = \frac{\text{\# of users of language } \ell \text{ in } G}{|G|}$$

for all 76 languages. The language distribution for all countries was also calculated, i.e.,

$$h_c(\ell) = \frac{\text{\# of users of language } \ell \text{ in country } c}{\text{\# of all users in country } c}$$

for all 143 countries. Using cosine similarity $g_G(\ell)$ is compared with $h_c(\ell)$ for all 143 countries. Let U_2 be the set of countries whose language distributions have a similarity score greater than threshold t_4. Finally, let $U_3 = U_1 \cap U_2$ that is, U_3 equals the set of countries common to both sets U_1 and U_2.

Recall that for each user group in the UTC offset dataset its location and therefore its country is known. In the above prediction if a user group's country belongs to U_i then the prediction is correct, false otherwise. Table III shows how the language distributions helps narrow down the list of possible countries for different thresholds t_4. In Table III the first three columns are the median number of countries associated with U_1, U_2, and U_3.

TABLE III
LANGUAGE HELPS CONSTRAIN THE SET OF POSSIBLE COUNTRIES FROM UTC WHILE IMPROVING PRECISION FOR HIGHER COSINE SIMILARITY, t_4

U_1	U_2	U_3	**P**	**R**	t_4
19	143	17	89.02	100.00	0.05
19	140	17	89.14	99.87	0.15
19	127	12	89.54	99.42	0.25
19	115	8	89.75	99.18	0.35
19	108	7	90.07	98.80	0.45
19	102	7	90.23	98.58	0.55
19	99	6	90.32	98.38	0.65
19	91	6	90.27	98.23	0.75
19	83	5	90.26	97.84	0.85
19	70	4	90.34	96.01	0.95

It can be seen that our final prediction, identified by the set U_3 is very accurate. It has very high Precision and Recall as well. From table recommend value of t_4 is from 0.85 to 0.95.

VII. APPLICATION ON INFLUENCERS-DATASET

The prediction methods, discussed in previous sections, are shown as a pipeline in Fig. 6. As an application the pipeline is applied over Influencers-Dataset. The goal is to identify geo-influencers and accurately associate them with the country that most of their followers are from. We are interested in high confidence influencer predictions because these in turn can be used as training data for an improved geocoder that will be presented in the following section.

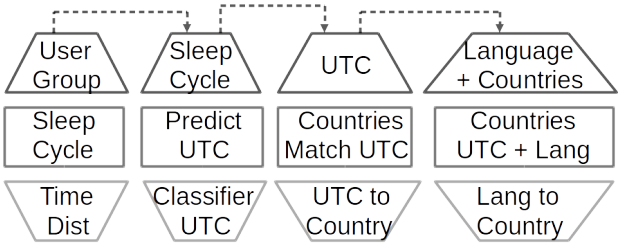

Fig. 6. Pipeline for predicting region of the world using time and language features. The top layer shows input, the middle layer output, and the bottom layer the name of the process. Each step sends its output as input to the next step.

One data point is the influencer's self-reported location. Another data point is the ratio of followers' self-reported locations belonging to a particular country. The expectation is that for geo-influencers the country associated with the self-reported location will match the most frequent country from influencer's followers. Self-reported locations that match a known GeoNames entry are utilized as described in Datasets section.

There were 100,712 influencers with a self-reported location that could be resolved using GeoNames. The country that the influencer associates themselves with is used as ground truth. For example, *@BBCNews* has a self-reported location 'London' which is associated with GBR using GeoNames. Ordering by the most frequent country from followers' self-reported locations, the top five countries and associated percent of self-reported locations are USA: 28.15%, NGA: 10%, IND: 6.3%:, GBR: 5.92%, and KEN: 4.81%.

A problem with self-reported location information from followers is that the labeled location focuses on the Latin alphabet with foreign city names appearing as they would be referred to by an English speaker (example Moscow instead of Moskva in Cyrillic). Therefore, English speaking users are incorporated more often than non-English speakers. Using the following steps we aim to test the different stages from Fig. 6 to help with this imperfect baseline:

1) Step 1a (S1a) – Get rid of influencers whose followers' time distribution do not exhibit a U-shaped parabola.
2) Step 1b (S1b) – Get rid of those influencers with $\sigma_G(\text{PST}) \geq t_3 = 0.49$.
3) Step 2 (S2_t_2) – UTC^P computed and used to obtain the set of possible countries U_1.

4) Step 3 (S3_t_4) – Reduce the set U_1 to set U_3 by constraining to countries that have cosine similarity above t_4.

Steps S1a and S1b should reduce the dataset to focus primarily on geo-influencers whose followers are concentrated in a single time zone. Steps S2_t_2 and S3_t_4 help identify and correct instances where the baseline is making poor predictions that do not match using countries based on time and time+lang features, respectively. If the top-ranked country from the baseline is within the set of countries it is returned as a prediction otherwise the second top country is used and so on. For example, baseline set for *@BBCNews* is [USA: 28.15%, NGA: 10%, IND: 6.3%:, GBR: 5.92%, and KEN: 4.81%], in which USA is the top prediction. But this does not match the set of possible countries from S2_t_2 and so NGA is used as it is the second best prediction; NGA has the same UTC as GBR and is thus within the set of possible countries using time features. Language features from S3_t_4 can further constrain the baseline to produce the expected result.

In the above, we have described a procedure that results in a set of possible countries making up U_3. We also consider a point estimate prediction where a single country with the best cosine similarity is returned for a narrow UTC range $t_2 = 0.25$ (this point estimate is denoted as S3_Point in Table IV).

Table IV shows how the baseline is constrained using each step of the pipeline. The second column shows precision across all influencers (100,712) and the fourth column shows precision across influencers not associated with USA (37,908).

TABLE IV
DIFFERENT STAGES OF PIPELINE IMPROVE BASELINE PRECISION

	All P	**Count**	**Foreign P**	**Count**
Baseline	86.34	100708	65.71	37904
S1a	86.23	95500	65.61	36087
S1b	89.98	71643	76.97	28725
S2_1	90.23	70544	77.39	28018
S2_0.5	90.95	65708	78.98	25845
S2_0.25	92.23	60387	79.43	20753
S2_1+S3_0.95	91.49	64940	78.15	23315
S2_0.5+S3_0.95	92.78	57584	80.86	19721
S2_0.25+S3_0.95	94.54	50978	80.94	13395
S2_0.25+S3_Point	97.2	35518	86.59	6587

Table IV illustrates that as time and language constraints are added the precision improves. Removing influencers that don't pass the sleep cycle test surprisingly doesn't help (row S1a), but getting rid of influencers with noisy PST predictions (row S1b) results in a big jump in precision (both measures are related to focussing on geo-influencers which are expected to have a strong follower concentration in a single geographic area). UTC offset information further constrains the baseline (rows S2_t_2). The highest precision is achieved when both UTC and language information are used.

VIII. IMPROVED GEOCODER

As already noted, the issue with the baseline Geocoder was that it handles only English based locations that have a match to GeoNames. Our proposed approach is to use the high confidence influencer predictions from previous section as training data for an improved geocoder. This geocoder will automatically learn common ways persons refer to locations in their native tongue.

country	1	2	3
USA	charlottenc	chicago	louisvilleky
IND	india	newdelhiindia	mumbaiindia
PRT	portoportugal	portugal	lisboaportugal
ITA	italia	milano	bolognaemiliaromagna
JPN	日本	東京	日本東京
RUS	Москва	Россия	СанктПетербург
THA	ประเทศไทย	กรุงเทพมหานครประเทศไทย	bangkokthailand
TUR	İstanbul	istanbul	bodrum
ISR	israel	ישראל	telaviv
UKR	ukraine	Украина	kyiv
DEU	berlindeutschland	hamburg	berlingermany
KOR	republicofkorea	seoulrepublicofkorea	대한민국서울
MEX	monterreynuevoleón	méxico	mexico
KEN	nairobi	nairobikenya	kenya
ROU	romania	bucureştiromânia	bucharestromania
AFG	afghanistan	kabulafghanistan	kabul
GRC	attikigreece	thessalonikigreece	ΑττικήΕλλάς
PRI	puertorico	puertoricousa	sanjuanpuertorico
COL	barranquillacolombia	bogotádccolombia	calicolombia
ESP	madrid	madridcomunidaddemadrid	españa

Fig. 7. Top three TF-IDF location features automatically learned from country documents. The benefit of this model is that it learns popular ways of referring to the country's locations in different languages and will include common phrases, abbreviations, and so on.

This new geocoder is based on a Term Frequency-Inverse Document Frequency (TF-IDF) model, where the country is the document and the terms are the self-reported locations of the influencer's followers. It is possible to apply this model because if the influencer and their followers belong to the same country, then the followers' locations will capture common ways persons refer to locations within that country.

TF-IDF vectors are generated using the Gensim package in Python[5]. From equation (3), the weights of TF-IDF are based on the frequency of term i in document j times log inverse ratio of total documents with the term over total D documents.

$$TF - IDF_{i,j} = frequency_{i,j} * log_2 \frac{D}{doc_{freq_i}} \quad (3)$$

Focusing on the top K TF-IDF features per country it is possible to verify that the model is generating reasonable vectors. Fig. 7 shows the top three features for several countries. These locations capture typical ways persons refer to locations within that country which can serve as useful features for geocoding type classifiers. The model also helps confirm that the countries with which the influencer's followers were associated with are indeed relevant.

Given a new influencer, the self-reported locations associated with the influencer's followers form a new document. A TF-IDF vector is built using self-reported location frequencies and the IDF component previously computed over the corpus of D documents. Cosine similarity is then used to return the country vector that is closest to the TF-IDF vector. Because the TF-IDF vectors may be very large in size we recommend utilizing Latent semantic analysis (LSA) to reduce dimensionality

[5]https://radimrehurek.com/gensim/

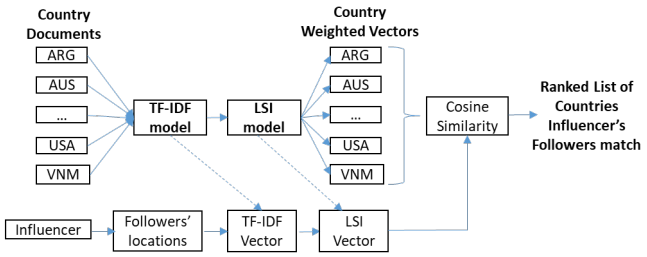

Fig. 8. The process by which TF-IDF model is learned from geo-influencers associated with a country and used for predicting other geo-influencers.

to at most 500 terms (from literature 50-500 is recommended as a standard [18]). Fig. 8 highlights the overall approach.

In Table V, the performance of the TF-IDF model is examined in the same way as was the baseline Geocoder in the previous section. Comparing Table IV and Table V we can see that the precision as well as number of predictions (count) improves across all values. This improved geocoder can be used to generate labels across additional influencers (that could not be predicted using the original baseline). The additional labels can be verified using time and language features and the TF-IDF model can be further improved. The process can repeat until the process converges, that is, when the TF-IDF model no longer improves.

TABLE V
TF-IDF MODEL PERFORMANCE FOR DIFFERENT STAGES OF PIPELINE

	All P	**Count**	**Foreign P**	**Count**
Baseline	88.44	100711	81.75	37907
S1a	88.36	95501	81.64	36088
S1b	92.53	71644	88.25	28726
S2_1	92.6	71282	88.33	28415
S2_0.5	92.97	69661	88.79	26946
S2_0.25	93.92	64021	89.53	21587
S2_1+S3_0.95	94.05	66635	89.89	24381
S2_0.5+S3_0.95	94.49	63109	89.99	21153
S2_0.25+S3_0.95	95.62	58024	91.1	16374
S2_0.25+S3_Point	97.54	35416	91.18	6624

IX. CONCLUSIONS

This paper has proposed a way of associating a user group with a geographic region using only their time and language features. The benefit of our approach is that these time and language features are universal and can be used to predict worldwide. Another benefit is that the classifier relies on features that 99% of all Twitter users have and the prediction can be made quickly since collecting this information does not require many Twitter API calls.

The drawback to our method is that it works at a high level for predicting regions of the world at the country level or larger. We envision that the features proposed will be utilized for augmenting other features as part of information fusion. For this reason, as an application, it was proposed to use these features for improving geocoding via a TF-IDF model. For our future work, we want to explore in more depth the influencers that the model did not classify due to time distribution that lacked a sleeping cycle. We believe that global vs. local influencers could be ranked using additional time-based features from these distributions.

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

REFERENCES

[1] Li, Yuchen, et al. "Influence maximization on social graphs: A survey." IEEE Transactions on Knowledge and Data Engineering (2018).
[2] Zheng, Xin, Jialong Han, and Aixin Sun. "A survey of location prediction on twitter." IEEE Transactions on Knowledge and Data Engineering 30.9 (2018): 1652-1671.
[3] Malik, Momin M., et al. "Population bias in geotagged tweets." People 1.3,759.710 (2015): 3-759.
[4] Graham, Mark, Scott A. Hale, and Devin Gaffney. "Where in the world are you? Geolocation and language identification in Twitter." The Professional Geographer 66.4 (2014): 568-578.
[5] Matci, Dilek Kk, and Uur Avdan. "Address standardization using the natural language process for improving geocoding results." Computers, Environment and Urban Systems (2018).
[6] Wong, Charlene A., et al. "Twitter sentiment predicts Affordable Care Act marketplace enrollment." Journal of medical Internet research 17.2 (2015).
[7] Stephens, Monica, and Ate Poorthuis. "Follow thy neighbor: Connecting the social and the spatial networks on Twitter." Computers, Environment and Urban Systems 53 (2015): 87-95.
[8] Panasyuk, Aleksey, Edmund Yu, and Kishan Mehrotra. "Improving Geocoding for City-level Locations." 2019 IEEE 13th International Conference on Semantic Computing (ICSC), IEEE, 2019.
[9] Compton, Ryan, David Jurgens, and David Allen. "Geotagging one hundred million twitter accounts with total variation minimization." Big Data (Big Data), 2014 IEEE International Conference on. IEEE, 2014.
[10] Cenni, Daniele, et al. "Twitter Vigilance: a multi-user platform for cross-domain Twitter data analytics, NLP and sentiment analysis." 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2017.
[11] Rejaie, Reza, et al. "Sizing up online social networks." Network, IEEE 24.5 (2010): 32-37.
[12] Meeder, Brendan, et al. "We know who you followed last summer: inferring social link creation times in twitter." Proceedings of the 20th international conference on World wide web. ACM, 2011.
[13] Inkpen, Diana, et al. "Location detection and disambiguation from twitter messages." Journal of Intelligent Information Systems 49.2 (2017): 237-253.
[14] Lau, Jey Han, et al. "End-to-end network for twitter geolocation prediction and hashing." arXiv preprint arXiv:1710.04802 (2017).
[15] Ebrahimi, Mohammad, et al. "A unified neural network model for geolocating twitter users." Proceedings of the 22nd Conference on Computational Natural Language Learning. 2018.
[16] Zannettou, Savvas, et al. "Disinformation Warfare: Understanding State-Sponsored Trolls on Twitter and Their Influence on the Web." arXiv preprint arXiv:1801.09288 (2018).
[17] Panasyuk, Aleksey, Edmund Yu, and Kishan Mehrotra. "Augmenting Google Search in Ranking Twitter Users." Semantic Computing (ICSC), 2019 IEEE 13th International Conference on. IEEE, 2019.
[18] Rehurek, Radim. "Subspace tracking for latent semantic analysis." European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2011.
[19] Efron B., R. J. Tibshirani. "An introduction to the bootstrap", Chapman & Hall;CRC 1998.