Analytical Workbench for Integrated Social Media Geo-Inference

Sanae Mahtal[1], Cristina Lupu[2], Benedikt Armbruster[1], Marvin Bechtold[1], Maximilian Reichel[1], Thomas Wangler[1], Dennis Thom[1], Steffen Koch[1], and Thomas Ertl[1]
[1]Institute for Visualisation and Interactive Systems (VIS), University of Stuttgart
[2]Alexandru Ioan Cuza University, Iasi Romania
[1]{Firstname.Lastname}@vis.uni-stuttgart.de
[2]lupuc_cristina@yahoo.com

***Abstract*—In the realm of social media monitoring and analysis, the availability of location-based information is of pivotal importance to understand the spatial behavior of social media users. Especially in fields like disaster management and urban planning, such data holds huge value for analysts and decision makers alike. However, as only few posts and messages in platforms like Twitter are already provided with GPS-coordinates or geo-tags by the users, researchers have proposed various algorithmic and model-driven means to infer this information from properties like the content, network, or geographic history of the users. Since many of these methods only focus on isolated features or specific models, this paper presents a comprehensive framework that allows to integrate, combine and compare multiple geo-inference schemes in a unified, standardized, and performance-optimized fashion.**

In addition to that, we present a visual interface, which offers an intuitive, real-time assessment of the accuracy of singular and combined methods as well as support in detecting and understanding possible anomalies. We demonstrate the usefulness and relevance of our approach in a comprehensive case study.

***Index Terms*—geo-inference, geo-prediction, visual analytics**

I. Introduction

With increasing popularity of social media, it has been investigated how data from services like Twitter or Facebook data can be utilized to gather semantic sensor information about critical real-world events. Various visual analytics researchers have demonstrated that such information can help to gain situation awareness in domains like disaster management, law enforcement, or epidemiology. For example, one of the earliest studies in the field, conducted by [1], demonstrated how Twitter data can be used to detect and assess the occurrence of natural disasters, such as earthquakes and typhoons, in real-time and with high spatial accuracy. Naturally, the availability of geographic information attached to user-generated content, also called Volunteered Geographic Information (VGI), is of particular relevance in such application areas. Only if messages can be pinpointed on a map, they start to become a useful data-source for distinguishing possible eyewitness reports from rumors, to highlight anomalous spatio-temporal patterns, and to correlate the data with other sources. So far, most studies in social media analysis have based their data on social media posts that were already tagged with accurate GPS locations or geo-tags (e.g. 'New York'). However, as estimated by [3], less than 1% of the 500 million daily Twitter messages are actually provided with such voluntary meta-information by their users. In order to increase that percentage, various researchers have therefore proposed algorithmic methods that infer the location of origin of messages or users, where such information is not given in advance. Based on different features of the data there already exists a broad range of such approaches for various scenarios and applications [4]–[6]. Most existing geo-inference techniques, however, still suffer from limited accuracy, limited scalability, and considerable location uncertainties. In addition to that, it has not been investigated how the existing methods could be cleverly combined with each other or how the associated uncertainties can be dealt with in a geographic information system (GIS).

The goal of this paper is to deliver a first building block towards these challenges by proposing an integrated workbench that enables combination, evaluation, and visualization of geo-inference techniques. Similar to well-known methods of ensemble learning, such as bagging, boosting, stacking, or custom defined functions, the workbench includes configurable decision-schemes to combine the inference methods, provide metrics to quantify the probability of errors, and enable insightful benchmarking of the results and uncertainties in a map-based visualization. It allows analysts to see the certainty of estimations, possible alternative locations, as well as explanations for the reasons of placement. Non-spatial diagrams can be used to highlight the reliability of information and illustrate how the amount of retrieved data can be increased if thresholds are decreased.

In the next section, we introduce related research. Based on our observations we then identify and discuss requirements for our integration approach in Section III. In Section IV, we describe the implementation techniques and detail the system components and visual interface. In Section V, we elaborate how this implementation is used in context of a practical use case. Section VI concludes and provides further reflections.

II. Twitter Location Guessing in a Nutshell

With roughly 300 million monthly active users, Twitter is one of the most popular social network platforms in the world[1]. The messages in Twitter, also called *tweets*, are short textual messages with a maximum length of 280 characters (140 until 2017), which can also contain images or links to other media. By default, tweets are shared publicly and can be accessed by anyone, even without a Twitter account. Tweets are a rich source of VGI as they combine the characteristics of social networking services with location-based information. The problem of predicting locations of social media data, where such information is not given, has been termed *geo-inference*, *geo-prediction*, or *geo-guessing*.

There are several geo-inference techniques which can be categorized by the features they are based on and by the algorithmic methods and models they use for prediction. In this paper we focus on the three most prominent location prediction methods, namely *content, context, and user-network*:

1) Content-based Guessers: The content of a tweet may already reveal information about the location of origin. Early approaches for content-based geo-inference have been provided by [7]. They introduced a concept of local words and used them to infer home locations of Twitter users. For example, a particular dialect may be common in a specific region or mentions of regional events. People from Madrid might talk more frequently about *Real Madrid* than people from other cities and we can use this information to determine the probability that a Tweet actually originates from there.

2) Context-based Guessers: A tweet is always connected to the author's Twitter profile. To predict the location of a tweet we may accumulate information from this profile like the self-declared home locations, websites and the time zone. It was reported by [9] that various of such indicators are an effective method to find the home locations of users. This was also leveraged by [10], who utilized the check-in history, past GPS tags, and other attributes for inference. In further works it was shown that timestamps and the time zone of different tweets posted by users can be used to predict their home location [4].

3) Network-based Guessers: Circles of friends with frequent interactions on social media platforms are often also friends in real-life sharing a common geographic origin. In accordance with Tobler's first law of geography [14], this means that the social interaction of Twitter users might not be independent from their geographic distance. Geo-location methods based on a Twitter network could exploit this inverse correlation between social interaction and the geographic distance. Several studies therefore considered following and mentioning actions in Twitter to predict the location of networked users.

4) Mixed methods for location prediction: In the recent past, researchers have also proposed combinations of the aforementioned approaches. For example, [12] proposed the UDI (unied discriminative influence) framework to combine message content and network location analysis in a unique model to profile users' home location. Furthermore, [13] proposed a method that learns associations from locations and keywords of earlier user messages to predict current ones.

In contrast to the existing works, this paper does not provide a new inference model. Instead, we focus on the practical need of an integrated system that allows the ad-hoc combination, evaluation, and visualization of existing approaches.

III. System Requirements

From our previous state of the art research, our previous projects in social media data analysis, as well as our initial assessment of the current data from Twitter, we have derived the following central requirements for our system design:

R1: The system shall allow easy, plug-in style integration of existing state-of-the-art geo-inference implementations. The user should be able to request separate and combined location guesses without having to understand the specific methods and features used in the process.

R2: The system shall provide an improved guess by "merging" two or more guesses based on different approaches. Combining the information given by two guessers can result in a more accurate geo-location or guide the user by providing insights about the uncertainties of the methods.

R3: The system shall distribute requests between different instances in the system to achieve higher performance. It should be possible to dynamically register new geo-guesser services to the back-end in an ad-hoc fashion.

R4: The system shall provide a graphical user interface that allows the user to query arbitrary sets of collected Twitter data based on their spatial and textual attributes. Based on this data, the user should be able to employ different location-inference schemes, which are then applied and evaluated. The result of this inference as well as the rates of error in terms of certainty and accuracy shall be represented using map and chart visualizations.

IV. Implementation

Our system is composed of a server-side *back-end*, where tweet-locations are inferred and merged (**R1** - **R3**) and a web-based *front-end*, which visualizes the guesses, indicates their accuracy, and gives additional information using an interactive map and charts (**R4**). Initially, the front-end is used to submit queries for specific test data and decide which location guessers and merge methods should be employed in the inference. In this section we will first give a brief overview of the technologies used in the implementation of our platform. We then give a detailed view of the system architecture and its components (Fig. 1) and will discuss methods of merging location guesses. Finally, we detail the methods for visualizing the results and associated accuracies.

Technologies and Data Structure

Our workbench is implemented in *Java* for the back-end, *Vaadin*[2] for the front-end, and it uses *MongoDB*[3] as the primary

[1] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] https://vaadin.com/

[3] https://www.mongodb.com/

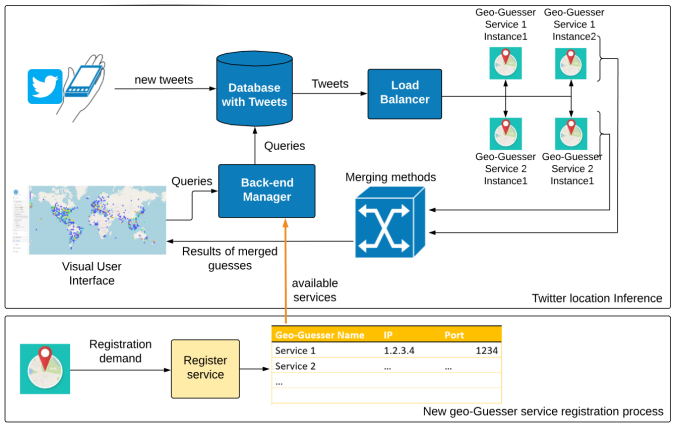

Fig. 1. Components and data flow in the back-end. The upper part depicts the registration of a new service and the lower part depicts the location inference of tweets.

database. The tweets are stored in MongoDB collections, such that each collection represents one day of tweets. We decided to use one collection per day because with a single collection for all tweets the indexes would get very big and the queries would be slower. A small amount of tweets in our collection is already provided with geo-locations by the users. These tweets can be used as ground-truth for benchmarking the guessers and merging schemes using our visualization. To enable fast retrieval of these tweets, they are registered in a 2D spatial index within MongoDB.

In order to ensure platform- and language-independence, we rely on *Protocol Buffers*[4] as data interchange format and we link the guesser services using *gRPC*[5] as communication mechanism. To this end, available geo-inference implementations in any programming language are enhanced with a gRPC client stub that acts as the interface to our gRPC server component in the back-end. In gRPC, client stubs for any programming language can be automatically generated from a service definition written in the Protocol Buffer language. This definition furthermore contains generic data structures that specify the composition of tweets and Twitter users as well as the expected results of a guesser. Generally, a guesser service response can contain one or multiple location guesses for a given tweet and associate each with GPS coordinates, certainties, and location precision.

System Overview and Components

Our back-end has four main functionalities. They comprise the storage and retrieval of previously collected tweets; the integration, management, and load-balancing of the different guesser service implementations; the methods to merge and combine guesser results; and the interfaces to communicate requests and responses from and to the front-end. An overview of the system can be seen in Figure 1. It houses the following primary functional components:

- **Back-end Manager** The back-end manager is the core component of the back-end containing two interfaces. One is an interface facing the services so they can register themselves. The second interface is for the front-end to send queries to the back-end and receive the possibly merged responses of the used services. The front-end can first request a list with all registered services. Based on this list the front-end can request guesses with specific services as well as combinations of services. The tweets to be guessed are specified by a MongoDB database query.
- **Register Service** Services can register themselves dynamically to the back-end manager via a gRPC interface. If a service starts the registration, the back-end tests the connection to the service first and accepts the registration. To register itself, the service sends a register request with the name of the service, the hostname or IP address and the port under which the service can be reached.
- **Load Balancer** The load balancer distributes tweets between redundant instances of the same service. Because the services often process tweets synchronously, the parallel requests to multiple instances of the same service result in a higher performance. The load balancer receives the tweets that should be guessed via gRPC interface from the manager, which has previously collected them from the database.

To test our implementation we have implemented gRPC client interfaces for two existing guessers, which are both defined in Java [4]. One of them is a content-based guesser, which estimates tweet locations based on toponyms, dialect, and location-specific terms in the content. The other one is a context-based guesser, which analyzes user-profiles to estimate their home location and most frequently visited locations. The latter can also be used to guess the location of individual tweets by estimating the most probable current user location.

Merging methods

Because the same tweet or user can be guessed by multiple approaches focusing on different features of the data, our merging approach allows a meaningful combination into a derived guess with improved accuracy. The merger component provides a method which takes the guessed tweet together with a map $\{(s_0, G_0), ..., (s_n, G_n)\}$, where s_i is the name of the service and $G_i = \{g_{i,0}, ..., g_{i,k}\}$ is the list with guesses from this service. Each guess $g_{i,j}$ is associated with geo-coordinates $lat(g_{i,j})$ and $lon(g_{i,j})$, a precision radius $rad(g_{i,j})$ in kilometers, and a certainty $crt(g_{i,j}) \in [0, 1]$. The latter indicates how likely the tweet originates from the area defined by loc and rad.

In our preliminary experiments, the following merge methods provided the most useful results and are thus implemented by default in our back-end:

- **Union** A straightforward approach is to simply take the guesses from all responses and unify them in a single list: $\bigcup_{i=1}^{n} G_i$. From this list, a visualization using the response would usually select the guess with highest certainty or allow multiple guesses for the same tweet to be displayed.

[4]https://developers.google.com/protocol-buffers/
[5]https://grpc.io

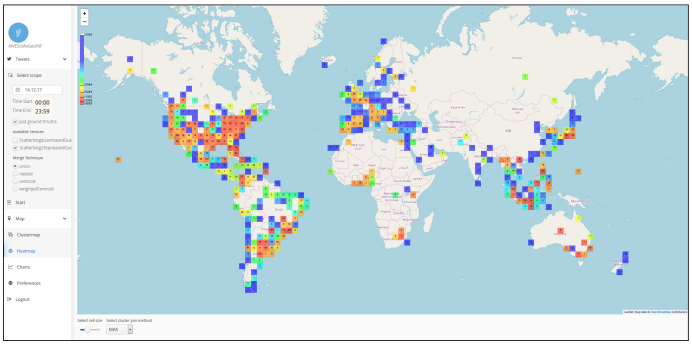

Fig. 2. Visual User Interface: Heatmap visualizing the error distances of tweet location guesses

- **Nested** The nested approach searches for guesses in an iterative fashion. The idea is to narrow the search area down from services that provide low precision but high certainty to services that provide high precision but low certainty. For example, one service might be good at finding the right country, the next one has good guesses for the city once the county is known, and the final service can determine precise locations within a given city. As the services usually output a list of guesses for different areas at once, we can apply this scheme directly to the results without having a means how the services can inform each other. The approach iterates through the results and uses the following function to calculate the guesses in the i-th iteration:

$$N_i = \begin{cases} G_0, \text{i=0} \\ \{g \in G_i | \exists g' \in N_{i-1} : \\ d(g,g') \leq rad(g')\}, i > 0 \end{cases} \quad (1)$$

  Here, the function $d(g, g')$ returns the distance between the coordinates of the guesses. Finally, the guesses contained in *N(n)* will be returned.
- **Centroid** The centroid approach selects from each service the guesses with the highest accuracy and uses these guesses to calculate a new combined guess. The most accurate guesses for each service are found using the following formula: $M_i = \{g \in G_i | \forall g' \in G_i - \{g\} : (crt(g) > crt(g')) \vee (crt(g) = crt(g') \wedge rad(g) \leq rad(g'))\}$. Finally, all of the best guesses are combined in $M = \bigcup_{i=1}^{n} M_i$ and we compute the weighted centroid g_c using the equations

$$lat(g_c) = \frac{\sum_{g \in M} lat(g) \cdot crt(g)}{\sum_{g \in M} crt(g)}$$

$$lon(g_c) = \frac{\sum_{g \in M} lon(g) \cdot crt(g)}{\sum_{g \in M} crt(g)}$$

Visual User Interface

The user interface (see Figure 2) allows the user to continuously evaluate the performance of individual guessers as well as to observe the results and possible anomalies that we get from merged approaches. The interface is comprised of a central map, an error chart view, as well as multiple widgets for querying archived tweet data and interacting with the visualizations. It comprises the following components:

- **Query Control** The initial query for tweets can be defined in terms of a temporal frame as well as a geographic bounding region which is selected directly on the map. The user can define whether the result should be restricted to ground-truth tweets or whether it should also contain tweets with inferred locations from previous runs. For tweets without ground truth data provided, the computation of the error distance is not possible. Therefore, without setting this flag, the heatmap and the accuracy chart (see below) are disabled. The services that should be used for the actual guessing of the tweet location can be selected from a list.
- **Clustermap** Once the back-end has retrieved the tweets and performed the selected guessing scheme, the computed locations will be visualized in a map of POI markers. Depending on the guessing scheme, the user can select whether only the best guess, the ground-truth, or all guesses for all tweets should be shown at a time. The cluster map adapts to the zoom level and combines near points in an area to aggregated markers showing a number of underlying locations. A user can click on a cluster to zoom to the respective area.
- **Heatmap** From the cluster map, the user can switch to the heatmap, which is the most important tool to indicate regional accuracy of the guessing as well as distinct local anomalies and outliers. To this end, the error distances are first computed as the Haversine distance from the ground-truth location of a tweet to the best guessed location. The heatmap then shows a uniform grid where the color of grid cells indicates the accumulated magnitude of error for tweets in the region. Based on the user-selection, the accumulation can be done with different methods, such as maximum, minimum, sum, or normalized average. This accumulated value v is finally mapped to a color hue or intensity using the formula $(\frac{v-v_{min}}{v_{max}-v_{min}})^{0.2}$ To ensure the comparability, the heatmap also shows a legend indicating the distribution of the colors.
- **Accuracy Chart** Finally, the user interface provides a chart that indicates the overall accuracy of the current guesses. Here, the x-axis shows an error distance and the y-axis indicates the percentage of guesses which are below that error distance. We used a mix of a logarithmic and linear scale for the distance, because in cities short deviations are more interesting, but in other areas it is only relevant to know if the tweet is located.

V. Use Case

To better understand the applicability and usefulness of our approach, we demonstrate how it can be employed in a practical use case. In our scenario, an analyst wants to check whether a specific location service based on content would combine well

with a service based on user context. For this analysis, the user would only retrieve ground-truth tweets to make error distance computation possible. She would furthermore restrict the data to a relevant geographic area and a recent time period to accurately reflect the current precision of the services. Finally, she would select the two guessers and the merging technique.

In the resulting cluster map visualization, the user would now already see some amount of error, as guesses shown outside the selected geographic scope are obviously wrong. For a more accurate representation, the user could switch to the heatmap to visualize the magnitude of error distances within the selected scope. From there it would be visible how bad the guessing behaved in general, but also whether there are specific regions within the scope where the performance is different than in others. This could have multiple causes, such as a smaller user base, different language use, or varying privacy awareness. If such outliers are found, the user might switch back to the cluster-map to investigate individual message contents in the region by selecting the markers. This might give her an idea about the reason for the different behavior. To validate initial hypotheses, the user would go back to the query phase and change the selection of guessers or merging methods. Based on her observations, she might either conclude that the merged approach is helpful or that one guesser is just superior for the relevant regions and that there is no need for a combination.

VI. Conclusion and outlook

In this paper we proposed a system to easily integrate, combine and visualize the output of multiple social media geo-inference services. With our tool it is possible to view the results of location-guessed tweets from different services on a clustered map. Additionally, a heatmap visualizes the error distances computed as the distance between the guess and the ground truth data. We also implemented different merge techniques to combine the results of different guessers. Our application can easily be expanded with additional geoinference services and more merging techniques. The back-end system is created in a modular fashion, so other applications are able to use the API as well. Currently we plan to include more guessers from available research projects to enable largescale benchmarking and combination. The results of this study will be published in our future work.

VII. Acknowledgment

This research was supported by DFG Priority Programme 'Volunteered Geographic Information: Interpretation, Visualisation and Social Computing' (DFG SPP 1894).

References

[1] Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." Proceedings of the 19th international conference on World wide web. ACM, 2010.

[2] Goodchild, M. F. "Citizens as voluntary sensors: spatial data infrastructure in the world of Web 2.0. IJSDIR 2: 2432." (2007).

[3] Sloan L, Morgan J, Housley W, Williams M, Edwards A, Burnap P, et al. (2013) Knowing the Tweeters: Deriving sociologically relevant demographics from Twitter. Sociological Research Online 18(3), article number: 7.

[4] Thom, Dennis, et al. "Using large scale aggregated knowledge for social media location discovery." 2014 47th Hawaii International Conference on System Sciences (HICSS). IEEE, 2014.

[5] Jurgens, David, et al. "Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice." ICWSM 15 (2015): 188-197.

[6] Han, Bo, Paul Cook, and Timothy Baldwin. "A stacking-based approach to twitter user geolocation prediction." Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2013.

[7] 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, 2010.

[8] Hecht, Brent, et al. "Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles." Proceedings of the SIGCHI conference on human factors in computing systems. ACM, 2011.

[9] Schulz, Axel, et al. "A Multi-Indicator Approach for Geolocalization of Tweets." Icwsm. 2013.

[10] Pontes, Tatiana, et al. "We know where you live: privacy characterization of foursquare behavior." Proceedings of the 2012 ACM conference on ubiquitous computing. ACM, 2012.

[11] 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.

[12] Li, Rui, et al. "Towards social user profiling: unified and discriminative influence model for inferring home locations." Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[13] Ikawa, Yohei, Miki Enoki, and Michiaki Tatsubori. "Location inference using microblog messages." Proceedings of the 21st International Conference on World Wide Web. ACM, 2012.

[14] Tobler, Waldo. "A computer movie simulating urban growth in the Detroit region." Economic Geography. Vol. 46. Taylor & Francis. 1970.