CitySense: Combining Geolocated Data for Urban Area Profiling

Danae Pla Karidi, Harry Nakos, Alexandros Efentakis, Yannis Stavrakas
IMIS, Athena RC
email: {danae, xnakos, efentakis, yannis}@imis.athena-innovation.gr

***Abstract*—Social networks, available open data and massive online APIs provide huge amounts of data about our surrounding location, especially for cities and urban areas. Unfortunately, most previous applications and research usually focused on one kind of data over the other, thus presenting a biased and partial view of each location in question, hence partially negating the benefits of such approaches. To remedy this, we developed the CitySense framework that simultaneously combines data from administrative sources (e.g., public agencies), massive Point of Interest APIs (Google Places, Foursquare) and social microblogs (Twitter) to provide a unified view of all available information about an urban area, in an intuitive and easy to use web-application platform. This work describes the engineering and design challenges of such an effort and how these different and divergent sources of information may be combined to provide an accurate and diverse visualization for our use-case, the urban area of Chicago, USA.**

Keywords- Social networks; Crowdsourcing; Open data; Geographic visualization.

I. INTRODUCTION AND MOTIVATION

The emergence of social networks, microblogging platforms, check-in applications and smartphone / Global Positioning System (GPS) devices in recent years has generated vast amounts of data regarding the location of users. To exploit this vastly growing data, recent research has focused on utilizing the geographic aspect of this information for statistical profiling of geographical areas [1], event detection, sentiment analysis of users, place-name disambiguation [2][3], identification of popular hotspots and their temporal variation, identifying and visualizing the typical movement pattern of users throughout the day [4][5], as well as improving existing city maps [6][7]. However, volunteered geographic information (VGI) contributed by online users is imprecise and inaccurate by design and it should, thus, be used with extra caution for critical applications.

Likewise, the increasing necessity for efficient location-based services and effective online advertising drove leading web providers (e.g., Google, Here, Bing, Foursquare) to store and offer Point of Interest (PoI) information to their users, usually through the use of online Application Programming Interfaces (APIs). Such an approach has several benefits, since the users not only have access to information about their nearby PoIs but they may also provide (or view) reviews or notify their friends of their current whereabouts. The same web services also allow shop-owners and enterprises to advertise their stores and the services they offer. However, as any commercial offering there are limitations on the use of those APIs, thus providing users with a very locally-limited view of the existing city infrastructure that cannot be directly used to extract additional information for city-scale areas.

On a separate front, the open data movement argued that citizens should have access to the data collected by government agencies, since they are the ones funding data collection through their taxes. A second strong supporting argument is that public access to government data helps individuals and enterprises to create apps that boost the economy and provide better services to the citizens, at no additional cost. Some countries and cities have openly released such data, which provide another alternative view of urban areas. Although this open data is official, curated, of excellent quality and impossible to collect by individuals, it has the obvious disadvantage that it cannot be real-time, it is usually not available through APIs and most importantly it may be updated at very infrequent intervals (e.g., census data), therefore at risk of being rather outdated.

Overall, the aforementioned three sources of information, i.e., volunteered geographic information, online PoI data and official open data each have their own strengths and weaknesses, regarding accuracy, update-rate, ease of use and availability. Likewise, applications or research that utilize and rely on only one of those types of data offer a biased and imprecise view of reality that could potentially be misleading. To remedy this, this work proposes the *CitySense* framework [1] that utilizes open data from administrative sources, online PoI APIs and social microblogs (tweets) to provide a unified view of our use-case, the urban area of Chicago. The main innovation and focus of the paper is to show how disparate datasets of various origins can be combined to provide a more complete picture of a geographical area. The corresponding web application [48] may be viewed with any modern web browser (Chrome, Firefox). Our emphasis is on how to efficiently spatially aggregate, visualize and present the end-user with an aesthetically pleasing and intuitive view of available raw data for any of these three sources, with minimum intervention, so that the end-user could freely interpret this information at his own will. As such, the CitySense application could be easily extended with additional features with minimal effort. Moreover, the CitySense Database is designed for the efficient storage and retrieval of the data acquired from the three above-mentioned sources. Overall, CitySense is a dynamic urban area viewer that integrates various datasets related to an urban area, providing a rich visualization of a city's life.

As a motivating example, consider a newcomer to the city, who has to search for a house in an unfamiliar area. She has to answer some questions, in order to narrow down and locate the neighborhoods to search. These questions may involve criteria like education facilities (''Where are the most popular

residential neighborhoods having high level educational facilities?'') and security (''Where is the downtown area with the lowest criminality measures?''). As another example, consider a tour operator that needs to track the tourist activity in a city, in order to offer improved tour packages and services. However, monitoring massive tourist activity using traditional methods would require lots of efforts, examination of many updating sources, hence huge costs and time involving off-line on-the-spot observation.

The central idea behind our approach is described in [1], where we presented the CitySense framework. In this paper, we extend that work by providing a deeper description of CityProfiler, the subsystem responsible for data collection, and a thorough description of the CitySense Database design and schema.

The outline of this work is as follows. Section II presents related work. Section III describes the objectives, the architecture and the web-based application of CitySense. Section IV describes the CitySense technical challenges. Section V describes the CitySense Database design. Finally, Section VI gives conclusions and directions for future work.

II. RELATED WORK

In recent years, as data from location sharing systems are constantly increasing, researchers have proposed a wide variety of "urban sensing" methods, based on location data derived from all kinds of sources: social media posts and check-ins, cellphone activity, taxicab records, demographic data, etc. Scientists combined social sciences, computer science and data mining tools, in order to derive useful knowledge regarding the life of cities. Cranshaw et al. [8] tried to reveal the dynamics of a city based on social media activity, while in [9][10], authors characterized sub-regions of cities by mining significant patterns extracted from geo-tagged tweets. Frias-Martinez et al. [11] focused on deriving land uses and points of interest in a specific urban area based on tweeting patterns and Noulas et al. [12] analyzed user check-in dynamics, to mine meaningful spatio-temporal patterns for urban spaces analysis. Much work has been done in the field of using social media textual and semantic content for urban analysis purposes. For example, Pozdnoukhov et al. [13] conducted real-time spatial analysis of the topical content of streaming tweets. Moreover, Noulas et al. [14] proposed the comparison of urban neighborhoods by using semantic information attached to places that people check in, while Kling et al. [15] applied a probabilistic topic model to obtain a decomposition of the stream of digital traces into a set of urban topics related to various activities of the citizens using Foursquare and Twitter data. Grabovitch-Zuyev et al. [16] studied the correlation between textual content and geospatial locations in tweets and Kamath et al. [17] used the spatio-temporal propagation of hashtags to characterize locations. Prediction methodologies have widely used geo-tagged social content. For example, Kinsella et al. [18] created language models of locations extracted from geotagged Twitter data, in order to predict the location of an individual tweet, in [19]-[22], the authors aimed to model friendship between users by analyzing their location trails and Cheng et al. [23] estimated a Twitter user's city-level location based purely on the content of the user's tweets. Moreover, researchers have focused on trend and event detection by detecting correlations between topics and locations [24][25]. Lately, many works have been published focusing on urban mobility patterns. For example, Veloso et al. [26] analyzed the taxicab trajectory records in Lisbon to explore the distribution relationship between pick-up locations and drop-off locations. In [27], the authors explored real-time analytical methodologies for spatio-temporal data of citizens' daily travel patterns in urban environment. The authors of [28]-[32] used the moving trajectory data of mobile phone users to study city dynamics and human mobility, while the authors of [33]-[36] analyzed the human mobility using social media data. Another field connected to urban analysis is the geodemographic classifications, which represent small area classifications that provide summary indicators of the social, economic and demographic characteristics of neighborhoods [37]. In the area of location demographics and socio-economic prediction and correlation, researchers have proposed a variety of methods based on geo-tagged social media data [38]-[40].

A wide variety of applications that describe the life of urban areas have been developed so far. For example, EvenTweet [41] is a framework to detect localized events in real-time from a Twitter stream and to track the evolution of such events over time. Moreover, the "One million Tweet Map" [42] is a web app that displays the last million tweets over the world map in real-time. Every second the map is updated, dropping twenty of the earliest tweets and plotting out the latest twenty keeping the number of tweets hovering at 1,000,000, showing clustered tweets in regions around the world, while users are able to zoom in or out on the map, and cause the re-aggregation of the clusters. Furthermore, the "tweepsmap" [43] application provides users with efficient geo-targeted twitter analytics and management and "trendsmap" [44] and "tweetmap" [45] shows the geo located latest trends from Twitter on a map. In Urban Census Demographics visualization field, the "Mapping America: Every City, Every Block" [46] enables users to browse local data from the Census Bureau's American Community Survey, based on samples from 2005 to 2009. Finally, "Social Explorer" [47] provides map based tools for visual exploration of demographic information, including the U.S. Census, American Community Survey, United Kingdom Census, Canadian Census, Eurostat, FBI Uniformed Crime Report, American election results, Religious Congregation Membership Study, World Development Indicators.

Although those works provide thorough insights in some aspects of life in an urban area, they fail to provide an integrated and global view of the city and to enable the user

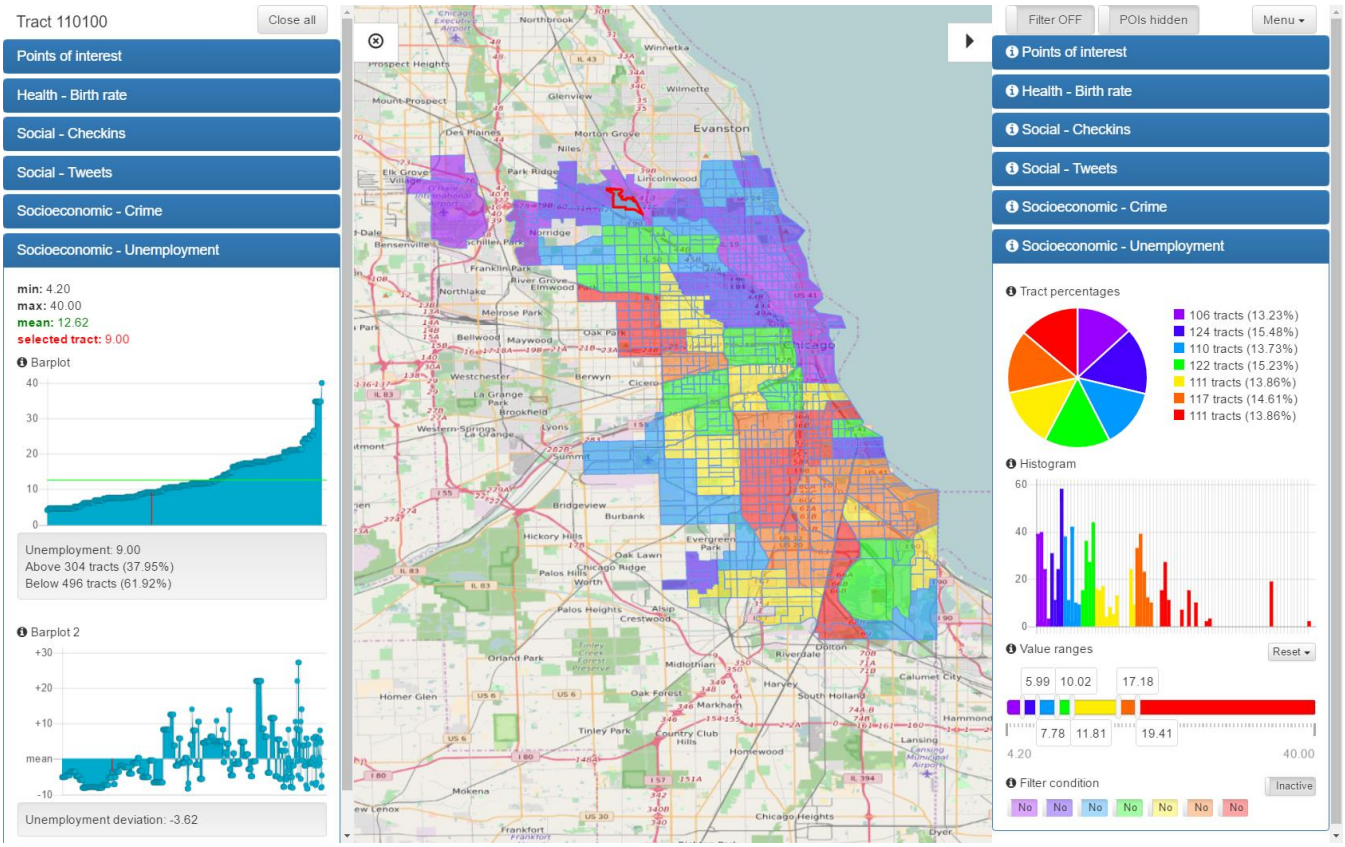

Figure 1. CitySense web-based user interface

to interactively answer questions by combining datasets. CitySense aims to fill these gaps by integrating multiple data sources and providing an interactive user interface supporting filters, multiple view options and drill down abilities.

III. BROWSING INTEGRATED CITY DATA

In this section, we present an overview of CitySense. We also discuss the objectives and present the features of the application.

A. Objectives and Architecture

CitySense is a dynamic urban area viewer, that integrates various datasets related to an urban area and provides a rich visualization of a city's life. The application can answer questions at many levels by exploiting the variety of datasets referring to a city and joining disparate data sources in an easy way. Users can view several aspects of city life statically or over time, for the whole city or for each part, mixing data sources to uncover patterns and information that would not be obvious from just observing the datasets.

The CitySense application [48] aims to provide a fast and easy way to:

- combine disparate data sources regarding various city aspects,
- filter data and drill down through a map-based visualization environment, and
- answer questions, explore and discover valuable information to convey the sense of the city.

The system architecture is presented in Figure 2 and includes the front-end Web-based Application of CitySense, the Data Infrastructure and Refresher units, the GeoServer that is discussed in Section IV-C and the CityProfiler subsystem (the dotted box in Figure 2) that was developed to collect the data related to the city from the data sources and is presented in detail in Section III-B.

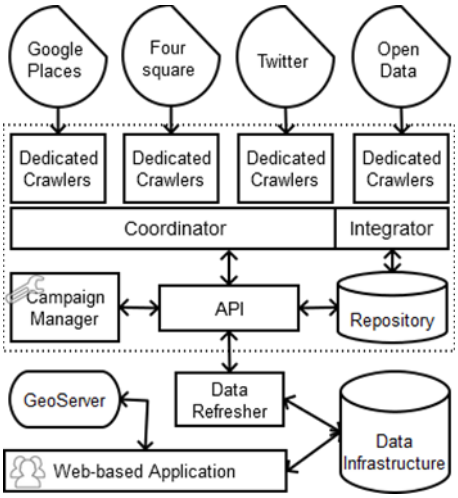

Figure 2. CitySense architecture

A screenshot of the CitySense web-based user interface is shown in Figure 1. The city of Chicago was selected for the pilot application, due to the amount and quality of official census data that are available. An additional reason is that Chicago's residents are exhibiting strong social media activity; moreover, a sufficient number of Points of Interest (PoIs) is available as well.

B. Harvesting Data with CityProfiler

CityProfiler (included in the dotted box in Figure 2) is a subsystem of CitySense, responsible for collecting data related to an urban area from diverse sources. Its basic functionality is to collect all available PoIs and tweets that come from the city and to store them in a repository together with relevant metadata.

Specifically, PoI data are extracted using Google Places and Foursquare APIs. Social Media data containing geospatial information are available via the Twitter API. The diversity of these sources raised the need for developing specific modules, called crawlers, to handle each data source. PoI crawlers collect PoI information using two methods. The first (general) method requires the selection of a geographic area and a PoI category, and returns a list of PoIs in the area belonging to this category. The second (special) method requires the selection of a PoI using a unique identifier and returns additional PoI information (name, address, phone number, opening hours, rating, etc.). In any case, the first (general) method returns the unique identifier of each PoI contained in the response list. This identifier can be used by the second (special) method to obtain more information about that PoI. The data obtained by the second method are stored in the repository.

In the case of the collection of geo-located tweets and check-ins, the corresponding crawler uses a method that requires the selection of a geographic area, and returns a list of geo-located tweets and check-ins, which have been posted from this area. Specifically, the crawler employs the Twitter Streaming API that provides real-time streaming data. Hence, the crawler has to collect geo-located tweets and check-ins dynamically. This is achieved by using the Twitter Streaming API in a sliding time window. Finally, the data obtained are stored in the repository.

Specifically, Google Places Crawler took 48 hours to complete the Chicago PoI collection. The 184,392 PoIs collected and stored in the database are depicted in Figure 3i. Meanwhile, the Foursquare Crawler had collected and stored 93,893 PoIs that are depicted in Figure 3ii. Figure 3iii depicts the complete PoI collection, from both Google Places and Foursquare Crawlers. Finally, Figure 3iv depicts the locations of 10,286 geo-located tweets (shown as blue dots) and 1,310 check-ins (shown as orange dots) that were collected by Social Media Crawler within these 48 hours.

CityProfiler provides an API and a GUI through which applications and users, respectively, can define and perform new collection campaigns. Each campaign, which is defined by certain parameters, results in an independent collection. These parameters control the individual crawlers that gather data through available APIs, and are the following:

- Crawling Duration: defines the duration of the campaign.
- Crawler Selection: selects which of the available crawlers (corresponding to distinct data sources like Foursquare, Google Maps, Facebook, Twitter, etc.) will participate in the campaign.
- Crawling Location: defines a crawling location by setting a point on the map and a range around it.
- Category Selection: selects target PoI categories and optionally keywords for the crawling to be based on. Keywords are used to narrow crawling, when the PoI category employed is deemed too broad (e.g., keyword "high school" is used when crawling Google Places for high schools, since "school" is the only applicable category). Category Selection can also collect all PoIs in a location, regardless of their category.
- Crawling Frequency Selection: some of the collected data need a systematic update, because of the changes that might occur to PoIs (e.g., a coffee shop might become a bar or new PoIs might show up). CityProfiler can perform repetitive campaigns with large duration in which multiple collections can be performed using the same parameters. Frequency Selection defines, therefore, how often the campaign should automatically restart.

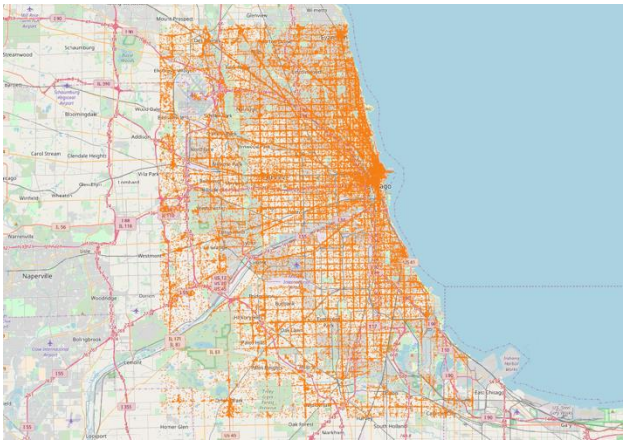

Figure 3i. Chicago Google Places PoIs

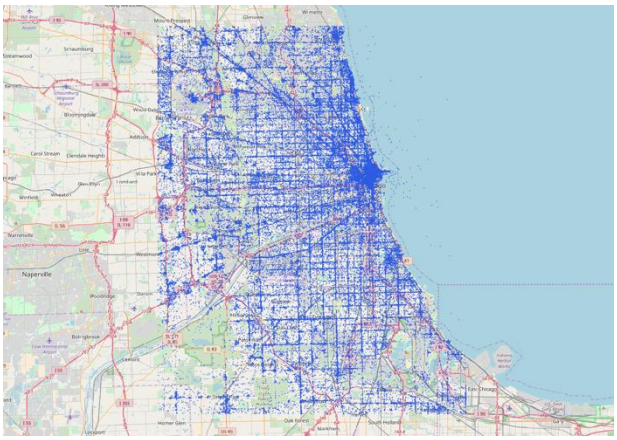

Figure 3ii. Chicago Foursquare PoIs

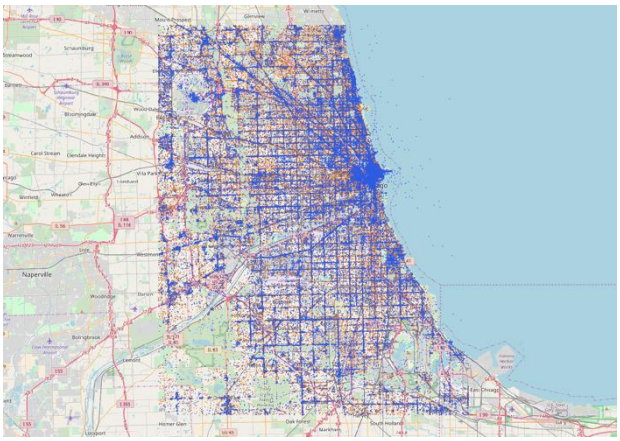

Figure 3iii. Chicago Google Places and Foursquare PoIs

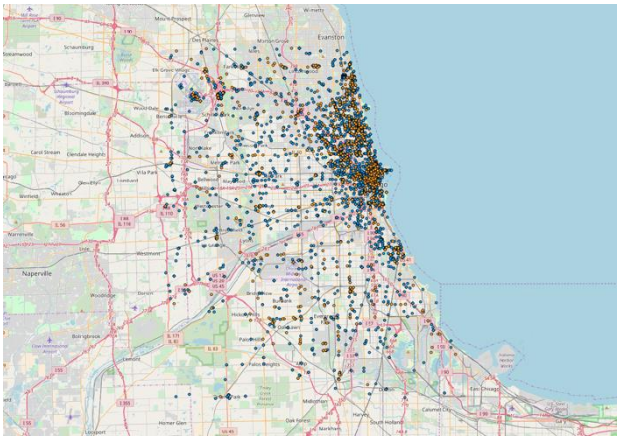

Figure 3iv. Geo-located tweets and check-ins locations in Chicago

CityProfiler is able to perform multiple campaigns in parallel, therefore there is a need of a Coordinator (see Figure 2) to control the crawlers and manage the campaigns. Moreover, CityProfiler manages resources in an intelligent way ensuring that all the restrictions imposed by the sources are met (e.g., maximum number of requests per time period), and that overlapping requests are avoided. Retrieved data are cleaned to exclude duplicates, and are temporarily stored in a repository.

C. Data Preprocessing and Integration

CitySense aims to shed light on the life of a city by exploiting three types of data: Points of Interest, Social Media and Open Census Data. PoI and Social Media Data are generated constantly by users and services. Therefore, we collect and update them in a regular and automatic way using CityProfiler, as discussed in Section III-B. Unlike these types of data, Open Census Data are generated by diverse sources (local authorities) at unpredictable time intervals. Moreover, they are published in various data formats (CSV, tab delimited, etc.). Therefore, Open Census Data require a case-dependent preprocessing and integration procedure keeping pace with their publication and taking into account the variety of data sources and formats. Finally, the diverse nature of these datasets requires a special integration regarding the aspect of time as well.

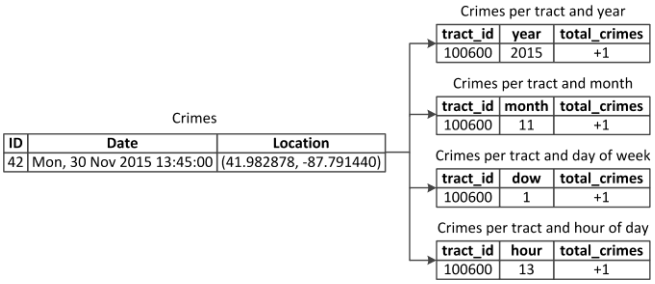

Figure 4. Crime data preprocessing and integration example

An example of the preprocessing and integration transformation regarding Crime data is presented in Figure 4. On the left side of the figure, we observe a single row of crime data that was downloaded in CSV format. This row represents a crime incident and contains its time and location. On the right side of the figure, we observe how this crime is represented in our database. Specifically, it is assigned to a tract (a specific geographical partition of the city) based on its location. The specific crime instance is represented by increasing the counter (total_crimes) in four tables, each representing a different time granularity: per year, month, day of week and hour of day.

D. CitySense Features and Design

Figure 1 shows CitySense web-based user interface. The central element of the visualization is the map of Chicago, which is divided in smaller sections, called tracts. Tracts are existing administrative divisions already used by the Chicago city government departments. Chicago contains 801 tracts and each of them describes a small area that is considered to be relatively uniform and corresponds ideally to about 1200 households (2000-4000 residents). Tract boundaries are always visible (blue line) on the map and when an individual tract is chosen its boundaries are highlighted with a red border line.

On the two sides of the map, CitySense provides two complementary views of Chicago. The first view appears on the right side and provides functions regarding the city as a whole. Hence, users can define visualization and filter options and observe the results both on the coloring of the city map and on distribution charts. The second view, is on the left side, and provides charts concerning only the selected tract, dark-highlighted on the map. This view, which appears when a tract is selected, helps users drill down to observe the special characteristics of each tract and to compare it with the city's overview. These views can be active concurrently, enabling users to observe different datasets in a general level and in tract level at the same time.

Both views provide visualizations and charts tailored to the corresponding dataset. For example, as shown in Figure 1, map coloring and charts visualize the Unemployment dataset.

To select a dataset, the user has to select a *data drawer*. Data drawers (dark rectangles) can be accessed concurrently in both views and represent the available datasets, e.g., "Points of Interest", "Health - Birth rate", "Social – Tweets", etc. According to the type of the particular dataset (see Section IV-A) each data drawer can contain different UI elements like pie charts, histograms, color range sliders and implement suitable functionality like value-based map coloring, temporal and combined filtering and superimposed PoI information.

The map coloring is based on user adjustable color range sliders that are available in each data drawer. Such a slider is presented in Figure 5 (top). After the color ranges are adjusted, users can define one or more colors as filtering parameters for combining various datasets. In other words, CitySense combines datasets (data drawers) by filtering the tracts based on their color. A color filtering slider, where only the violet color (leftmost) is defined as filtering condition, is shown in Figure 5 (bottom).

The tracts that satisfy the conditions set in all data drawers are colored grey on the map. Figure 6 shows the filter output for Social-Tweets and Socioeconomic-Crime datasets.

Certain datasets are visualized based on temporal aspects (per month/day/hour). The temporal functions described here are shown in Figure 7. Thus, users can select the time granularity, e.g., month of year, day of week, hour of day to adjust the charts and map coloring accordingly. Additionally, users can color the map or view the tract charts based on a specific month, day, or two-hour interval.

Finally, the CitySense application enables the user to see superimposed PoI information on the map at any moment. The user can select one or more categories (Food, Residence, Outdoors & Recreation, etc.) and the corresponding PoIs appear on the map as shown in Figure 8.

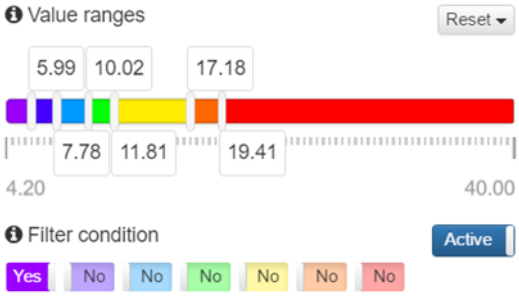

Figure 5. Coloring and filtering color slider

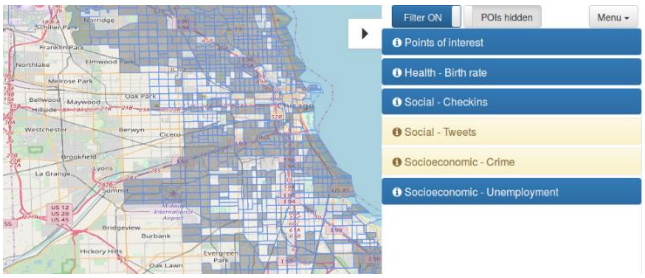

Figure 6. Filtered map

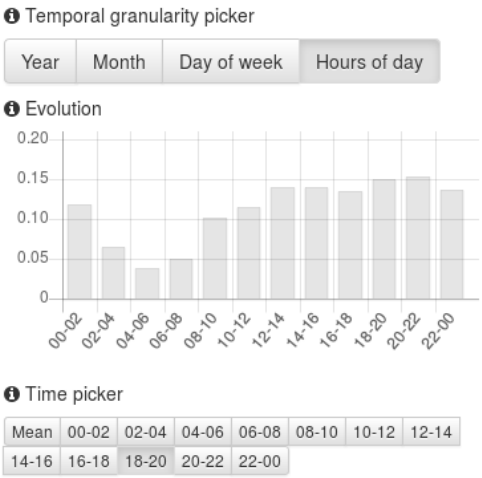

Figure 7. Temporal pickers

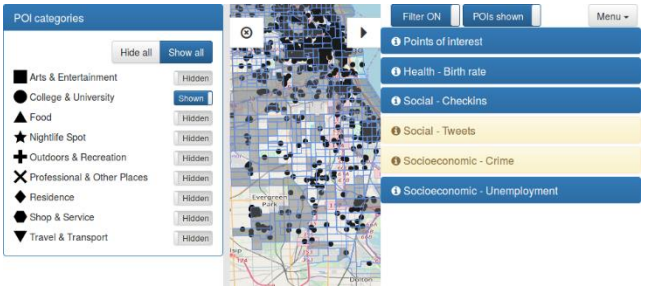

Figure 8. Filtered map with PoIs

IV. TECHNICAL CHALLENGES

In this section, we present in detail the technical challenges of the CitySense application.

A. Organizing Disparate Datasets

In order to convey the sense of a city CitySense must integrate and visualize a variety of datasets. The data sources that are integrated consist of demographic, social media and PoI data. The diverse nature of these datasets requires a different integration manipulation regarding the aspect of time. As we show in Table 1:

- Open Census Data can be visualized both in a static (overtime) or in a temporal way (per month/ day/hour). For instance, Health and Unemployment data are visualized statically and Crime data temporally.
- Social Media Data can be visualized in a static, temporal or dynamic way, although they are produced and gathered dynamically (real time). The feature of real time dynamic visualization of social media data is currently being developed.
- Point of Interest Data are visualized in a static way.

TABLE I. DIVERSITY OF DATASET VISUALIZATION REGARDING TIME

	static	temporal	dynamic
Open Census Data	✓	✓	
Social Media Data	✓	✓	✓ ongoing development
Point of Interest Data	✓		

The above organization of data helped to overcome their diversity and provide coherent visualization and treatment within the application.

A related problem is that of the initialization of the user-adjustable color range sliders. Our goal was to provide a reasonable use of map coloring to help users draw conclusions about the city. Therefore, we provided two options for initialization. The first, the value-based initialization option, breaks the slider based on equidistant values. However, this approach is sensitive to data with extreme outlier values or extreme concentration in certain ranges. The second option provides a percentage-based initialization, hence breaks the slider based on equal distribution percentages. However, this approach is sensitive to having many tracts with almost equal values. As an example, Figure 9 shows the value-based initialization for crime data.

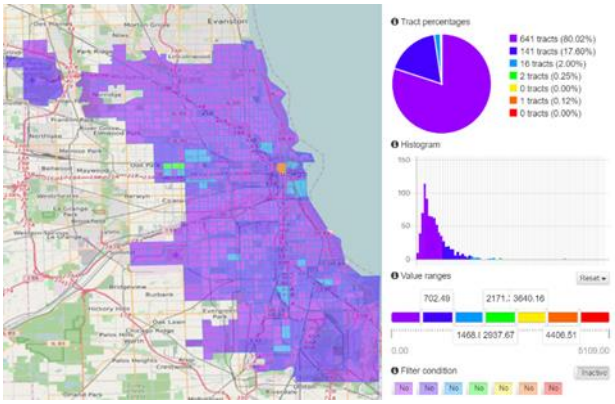

Figure 9. Value-based initialization

As we can observe in the histogram shown in Figure 8 (right), the crime data mainly occupy a small value range, between 0 and 1468, resulting in the almost two-colored map (violet and indigo – colors may not be visible on printed document) of Figure 9 (left). To address this issue, we use the percentage-based initialization, which is presented in Figure 10.

The resulting map coloring shown in Figure 10 (left) is obviously improved. However, as we can observe in the tract percentages shown in Figure 10 (right), the crime data distributions are not equally divided, because some tracts have almost equal values with respect to the range step and, therefore, cannot be equally classified.

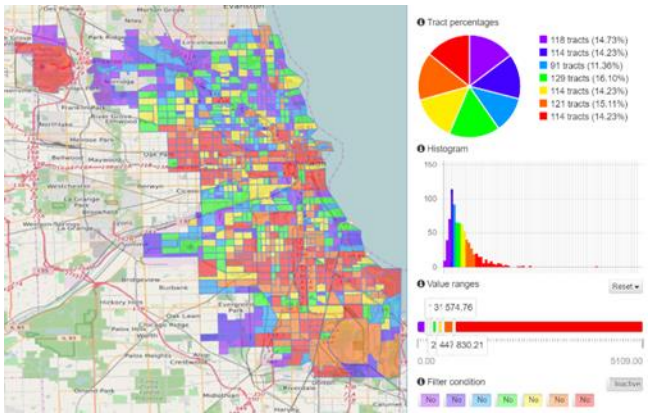

Figure 10. Percentage-based initialization

B. Acquiring Data of an Area

CityProfiler gathers PoI data from an urban area by performing calls to API services like Google Places and Foursquare, which set restrictions and constraints. A naïve crawling of PoIs, in terms of a whole city, would not be able to collect the entire amount of PoIs, but only a small portion of it as dictated by the rules imposed by the source. CitySense deals with this issue by breaking the area to smaller parts in advance. Specifically, the city is divided in squares of longitude and latitude of 0.03 degrees before the PoI crawling. In case this method doesn't gather all the PoIs, then recursion is used.

Additionally, CityProfiler collects real-time social data from the city. In order to achieve this, CityProfiler performs a real-time crawling of tweets with Twitter Streaming API, using a location box, which encompasses the city as filter parameter. Only the geo-located tweets (that are posted along with their latitude and longitude) are collected. In order to collect social check-ins and the PoIs that they were posted at, CityProfiler performs a call to Foursquare API every time a tweet contains a Swarm (mobile app that allows users to share their location within their social network) link. This way, the application collects temporal information concerning geo-located tweets, including their hashtags and check-ins posted at city PoIs.

C. Implementation and Efficiency Issues

Several implementation decisions had to be made, so that the application would run efficiently. The application needed to be lightweight with respect to memory and processing power consumption, as well as responsive with respect to the end-user experience.

At certain parts of the application a large number of geometries, namely tens of thousands of PoIs, needs to appear on the screen simultaneously. The option of handling each geometry as a separate entity and drawing it on the map separately would require much memory and processing power especially when zooming in and out the map. The approach employed is based on drawing relevant geometries as one image layer containing all geometries. GeoServer

(shown in Figure 2) is leveraged for generating and serving image layers. For additional efficiency, the built-in caching functionality of image layers by GeoServer is utilized. This way subsequent requests may use already generated image layers.

The application's requirements involve aggregate queries on data, spanning the geospatial and temporal dimensions. Such queries take much time, if performed on raw data, resulting in degradation of responsiveness for the end-user. In order to avoid costly operations during runtime, a preprocessing stage is employed. The database design for preprocessed data was driven by the critical use cases available to the end-user via the UI. As an example, the user is able to query for check-in data, aggregated per tract, pertaining to a specific PoI category and a specific day of week. Raw check-in data contain the geographic coordinates of the PoI, the category of the PoI, as well as the date and time of the check-in, across two tables. Tract geometries are stored in a separate table as well. Such a query cannot be executed instantaneously. During the preprocessing stage, the coordinates of the PoIs are mapped to the intersecting tracts, the days of week are extracted from date and time, and aggregation per tract and day of week is performed. The preprocessing results are stored in database tables. This way efficient querying for check-ins, in a specific PoI category, on a specific day of week, is achieved. Separate tables are employed to deal with different time granularity aspects of the temporal dimension, i.e., there exist separate tables for years, months, days of week, hours of day. Another optimization measure in the same direction is the delegation of heavy computations to the initialization stage of application services. This has an effect on the start-up time of the application, but speeds up requests during runtime.

The application currently encompasses a relatively small number of datasets, so data handling is manageable using PostgreSQL database system, as described in Section V. If the datasets grow in number, a data warehouse can be used to facilitate data management and efficient processing of aggregate queries.

D. Adapting to Other Cities

One of our primary concerns during the development of the CitySense framework was adaptability of the framework to other cities. Adaptation of CitySense to another city is comprised of three major tasks, partitioning of the city area, integration of Open Census Data and implementation of the relevant access methods, and specialization of the front-end according to the available city data.

1) City Area Partitioning

CitySense is essentially parametric with respect to the attributes that define the city of interest, namely a bounding rectangle that encloses the city and a partitioning scheme for the city. The partitioning scheme may in theory consist of an arbitrary set of polygons that collectively cover the whole city. Choosing a partitioning scheme is, nevertheless, not that straightforward. In order to effectively choose a partitioning scheme, official administrative partitioning schemes should be looked into (e.g., community areas, ZIP codes, census tracts), focusing on partitioning schemes used in Open Census Data of interest. Disregarding such partitioning schemes and employing an arbitrary one could result in Open Census Data of interest rendered either useless or hard to map to the employed partitioning scheme. Should the official partitioning scheme be considered too fine-grained, grouping could be applied to the small partitions, in order to acquire a more coarse-grained partitioning scheme to use. Should the official partitioning scheme be too coarse-grained, segmentation of the large partitions into smaller ones would result in a more fine-grained partitioning scheme to use.

2) Open Census Data Integration and Access

Open Census Data is the most cumbersome type of data to integrate into CitySense. While CityProfiler data are the same, irrespective of the city of interest, Open Census Data could be vastly different, even among different types of Open Census Data for the same city. Open Census Data could be stored in database tables or files. As long as data transfer from the back-end to the front-end is of the same form, regardless of the type of data, all underlying implementation details have no other constraints. Open Census Data will often make use of a specific partitioning scheme that will generally diverge from the partitioning scheme applied to the city. Such data will need to be mapped to the employed partitioning scheme. There is no recipe for universally handling this issue, hence the aforementioned suggestion to let Open Census Data drive the choice of a partitioning scheme for the city. Open Census Data with temporal and/or categorical dimensions should be stored in a way that will facilitate efficient data retrieval based on corresponding parameters. The methods that implement data access should also support temporal and/or categorical parameters, if should such dimensions exist for a specific type of Open Census Data. While parameters are specific to each type of Open Census Data, the response from the back-end should always be of the same form, so that all response data can be treated uniformly by the front-end.

3) Front-end Specialization

Specialization of the front-end in order to support the city data available by the back-end is the final task in the process of CitySense adaptation. Each dataset is represented by a data drawer both in the left and the right sidebar. All datasets follow the same protocol with respect to the data sent by the back-end. The only thing that needs to be specialized per dataset is the data picker, in case that one exists for a specific dataset. The data picker is used to navigate categorically and/or temporally within the dataset. The data picker parameters will be transformed to request parameters that are received by the back-end. The back-end response will follow

the data transfer protocol. The data drawer, therefore, needs no other specialization before it can display the received data.

E. Linear Prediction Model

Very often the datasets are not independent of each other. For example, infant mortality is very likely to be income-related, and is increased in areas with low income. One way to predict values of a variable (response) based on the corresponding values of other variables (predictors) is to find a suitable linear model based on the method of least squares. There are two reasons for constructing such models:

- They can provide an "exploratory analysis" of data. Through comparing the predicted values with the actual it is possible that correlations between variables can be explored, e.g., crimes are associated with income and unemployment.
- They can provide an estimation of a missing value for a tract, since this value can be inferred based on the values of predictors for this tract.

The CitySense application supports the construction of linear models for any of the available datasets. As an example, we consider crime data. From the application menu, we can create linear models (select "New model fit"), regarding crime as response and any combination of predictors. As an example, consider Crime as response, with predictors the Income, the Unemployment, the Checkins and the Points of Interest. The result of the model (the prediction for the crime values), which are shown in Figure 11, when compared with the actual data for crime, confirms the association of crime with the specific predictors.

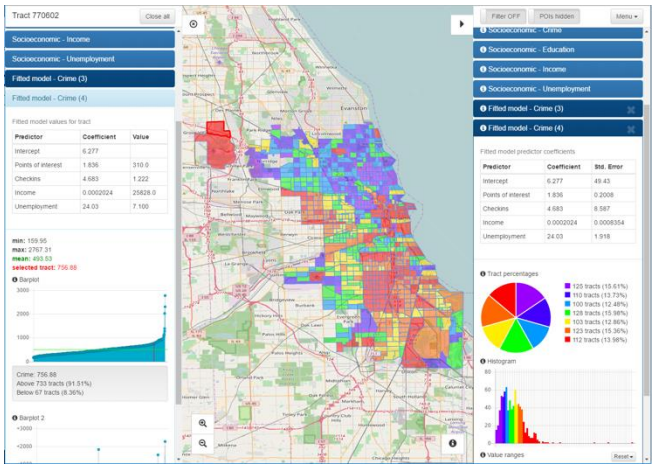

Figure 11. Linear model visualization

Moreover, before the model construction, there was no crime value for the upper left tract in the dataset. As we observe in Figure 11, the same tract has a value and appears in red color. Since this tract is selected (red outline), the data for that tract as derived from the linear model are shown in the left tray.

V. CitySense Database

The diverse and complex nature of the data used in CitySense application poses considerable challenges in data handling and storage. Thus, data come in many different formats and comprise varying content. Therefore, CitySense Database was designed with a flexible structure that can accommodate the diverse styles of the data.

A. Database Requirements

In order to meet the requirements of the application, we opted to use a relational database, which allows the application to immediately retrieve the necessary information based on several criteria. The efficient organization of information regarding PoI, Open Census, and Social Media data, requires a database that can adapt to their diverse nature. Specifically:

- Points of Interest require the storage of accompanying features such as name and location. The tract where the PoI is located is also stored, in order to achieve efficient aggregation based on tracts. At the same time, it is necessary to store the PoI category that each PoI belongs to (Arts & Entertainment, Food, etc.), so that aggregation based on categories is possible as well. Finally, Points of Interest have no need for separate time information, since they are considered static in time.
- Open Census Data comprise both static and temporal data. For example, demographic data, socio-economic indicators, and health indicators are presented as static data and, therefore, temporal information storage is not required for them. On the other hand, crime data require the storage of information about crime distribution over time (per month, day of week, etc.).
- Social Media Data, namely tweets and check-ins, are essentially temporal data. It is, therefore, necessary to store the information on their distribution over time. Besides spatial information storage, which is necessary for both tweets and check-ins, our system also needs to store the associated PoI category for check-ins. This is needed for check-in aggregation per PoI category.

B. Database Design and Schema

The main goal of the CitySense project is to exploit as much data as possible for a particular area, in order to have a realistic depiction of its "trace" on multiple levels. Therefore, the database presented here is designed to meet all the requirements of CitySense application, and also to be flexible to adapt to future requirements and store new data that may arise. The corresponding ER diagram of the database is presented in Figure 12.

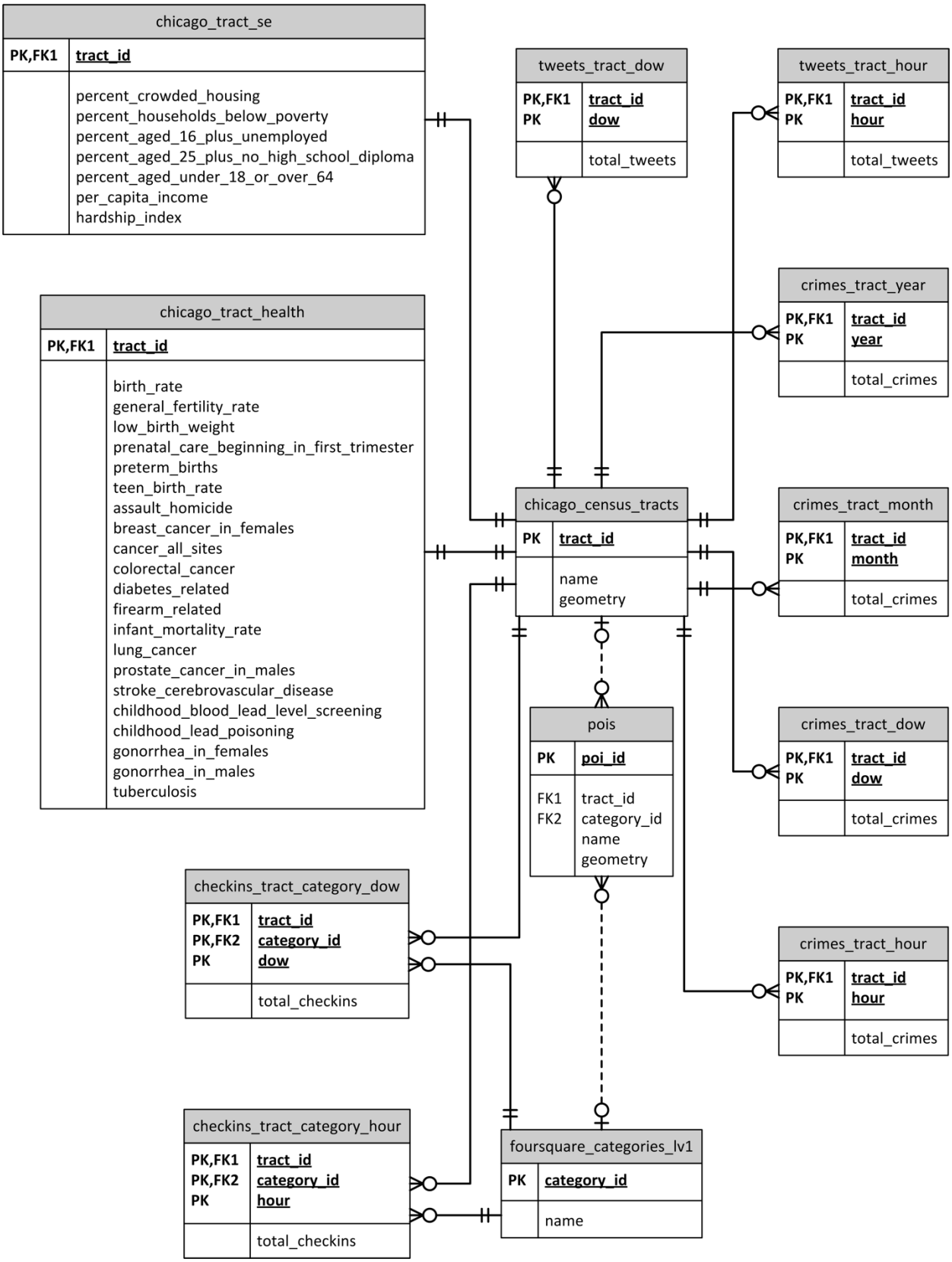

Figure 12. Database schema

The following is a detailed description of all the tables forming the database:

- **chicago_census_tracts:** The table contains all Chicago tracts that the application supports. More specifically, it contains the tract unique identifier, tract name, and polygon geometry that encompasses the tract.
- **chicago_tract_se:** The table is used to store the socio-economic indicator data used by the application. More specifically, it contains the per capita income, poverty rates, unemployment rate, etc., per tract.
- **chicago_tract_health:** The table is used to store the health indicator data used by the application. More specifically, it contains indicators of infant mortality, premature births, fertility, etc., per tract.
- **crimes_tract_year:** The table contains the crime data that the application uses. The table stores the number of crimes per tract at a yearly time breakdown.
- **crimes_tract_month:** The table contains the crime data that the application uses. The table stores the number of crimes per tract at a monthly time breakdown.
- **crimes_tract_dow:** The table contains the crime data that the application uses. The table stores the number of crimes per tract and day of week.
- **crimes_tract_hour:** The table contains the crime data that the application uses. The table stores the number of crimes per tract and time of day.
- **tweets_tract_dow:** The table contains the geo-located tweets data that the application uses. The table stores the number of tweets per tract and day of week.
- **tweets_tract_hour:** The table contains the geo-located tweets data that the application uses. The table stores the number of tweets per tract and time of day.
- **foursquare_categories_lv1:** The table is used to store the PoI categories that the application uses. The information stored consists of the unique identifier of the PoI category and the category name.
- **pois:** The table is used to store the Points of Interest that the application uses. The information stored consists of the unique identifier of the point of interest, its name, the point geometry that pinpoints the PoI's location, as well as its category and the tract in which the PoI is located.
- **checkins_tract_category_dow:** The table contains the data of the geolocated tweets that represent check-ins at some point of interest. The table stores the number of check-ins per tract, PoI category, and day of week.
- **checkins_tract_category_hour:** The table contains the data of the geolocated tweets that represent check-ins at some point of interest. The table stores the number of check-ins per tract, PoI category, and time of day.

The application is using a PostgreSQL database system. PostgreSQL is an open source relational database management system. To manage spatial information in an efficient way, we used the PostGIS extension of PostgreSQL, the official spatial extension of PostgreSQL. PostGIS is a software library that adds support for geographic objects (polygons, points) to PostgreSQL databases.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented CitySense, a dynamic urban area viewer that provides a rich visualization of city's life, by integrating disparate datasets. The application helps answer questions and reveals several aspects of city life that would not be obvious from just observing the datasets. In order to accomplish that, we developed special data collection and managing tools, rich visualization and filtering functions and dealt with several technical challenges. Currently, we are developing the feature of dynamic visualization of social media data (tweet posts, check-ins and hashtags). The support for dynamic datasets could be used to cover city power consumption and traffic data in the future. Another future target concerns the incorporation of road network information into our system. Users could calculate the actual distance between PoIs, by exploiting special road network based functions provided by CitySense. Finally, as more and more data is integrated through CitySense, the problem of scalability will arise. Therefore, a cloud data infrastructure is considered to fit CitySense's future data storing and managing needs.

ACKNOWLEDGMENT

This work was partially supported by the "Research Programs for Excellence 2014-2016 – CitySense".

REFERENCES

[1] D. Pla Karidi, H. Nakos, A. Efentakis, and Y. Stavrakas, "CitySense: Retrieving, Visualizing and Combining Datasets on Urban Areas," Proceedings of the the Ninth International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA 2017.

[2] A. Crooks, D. Pfoser, A. Jenkins, A. Croitoru, A. Stefanidis, D. Smith, S. Karagiorgou, A. Efentakis, and G. Lamprianidis, "Crowdsourcing urban form and function," International Journal of Geographical Information Science 29.5, pp. 720-741, 2015.

[3] E. Drymonas, A. Efentakis, and D. Pfoser, "Opinion mapping travelblogs," In Proceedings of Terra Cognita workshop in conjunction with the 10th International Semantic Web Conference 2011, pp. 23-36.

[4] A. Efentakis, S. Brakatsoulas, N. Grivas, G. Lamprianidis, K. Patroumpas., and D. Pfoser, "Towards a flexible and scalable fleet management service," In Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science, p. 79, 2013.

[5] A. Efentakis, N. Grivas, G. Lamprianidis, G. Magenschab, and D. Pfoser, "Isochrones, traffic and DEMOgraphics," In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 548-551, 2013.

[6] A. Efentakis, S. Brakatsoulas, N. Grivas, and D. Pfoser, "Crowdsourcing turning restrictions for OpenStreetMap," In EDBT/ICDT Workshops, pp. 355-362, 2014.

[7] A. Efentakis, N. Grivas, D. Pfoser, and Y. Vassiliou, "Crowdsourcing turning-restrictions from map-matched trajectories," Information Systems, 64, pp. 221-236, 2017.

[8] J. Cranshaw, R. Schwartz, J. Hong, and N. Sadeh, "The livehoods project: Utilizing social media to understand the dynamics of a city," Association for the Advancement of Artificial Intelligence, 2012.

[9] S. Wakamiya, R. Lee, and K. Sumiya, "Crowd-sourced urban life monitoring: urban area characterization based crowd behavioral patterns from twitter," In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, p. 26, 2012.

[10] R. Lee, S. Wakamiya, and K. Sumiya, "Urban area characterization based on crowd behavioral lifelogs over Twitter," Personal and ubiquitous computing, 17(4), pp. 605-620, 2013.

[11] V. Frias-Martinez, V. Soto, H. Hohwald, and E. Frias-Martinez, "Characterizing urban landscapes using geolocated tweets," International Confernece on Social Computing (SocialCom), pp. 239-248, 2012.

[12] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An Empirical Study of Geographic User Activity Patterns in Foursquare," International Conference on Web And Social Media (ICWSM), pp. 70-573, 2012.

[13] A. Pozdnoukhov and C. Kaiser, "Space-time dynamics of topics in streaming text," In Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks, pp. 1-8, 2011.

[14] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks," The Social Mobile Web, 11(2), 2011.

[15] F. Kling and A. Pozdnoukhov, "When a city tells a story: urban topic analysis," In Proceedings of the 20th International Conference On Advances in Geographic Information Systems, pp. 482-485, 2012.

[16] I. Grabovitch-Zuyev, Y. Kanza, E. Kravi, and B. Pat, "On the correlation between textual content and geospatial locations in microblogs," In Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data, p. 3, 2014.

[17] K. Kamath., J. Caverlee, K. Lee, and Z. Cheng, "Spatio-temporal dynamics of online memes: a study of geo-tagged tweets," In Proceedings of the 22nd International Conference on World Wide Web, pp. 667-678, 2013.

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

[19] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the gap between physical location and online social networks," In Proceedings of the 12th ACM International Conference On Ubiquitous Computing, pp. 119-128, 2010.

[20] D. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, "Inferring social ties from geographic coincidences," Proceedings of the National Academy of Sciences, 107(52), 2010.

[21] E. Cho, S. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082-1090, 2011.

[22] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," In Proceedings of the 19th International Conference on World Wide Web, pp. 61-70, 2010.

[23] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 759-768, 2010.

[24] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi, "Geoscope: Online detection of geo-correlated information trends in social networks," Proceedings of the VLDB Endowment, 7(4), pp. 229-240, 2013.

[25] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection," In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, pp. 1-10 ,2010.

[26] M. Veloso, S. Phithakkitnukoon, and C. Bento, "Sensing urban mobility with taxi flow," In Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks, pp. 41-44, 2011.

[27] L. Liu, A. Biderman, and C. Ratti, "Urban mobility landscape: Real time monitoring of urban mobility patterns," In Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management, pp. 1-16, 2009.

[28] C. Ratti, D. Frenchman, R. Pulselli, and S. Williams, "Mobile landscapes: using location data from cell phones for urban analysis," Environment and Planning B: Planning and Design, 33(5), pp. 727-748, 2006.

[29] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti, "Cellular census: Explorations in urban data collection," IEEE Pervasive Computing, 6(3), 2007.

[30] M. Gonzalez, C. Hidalgo, and A. Barabasi, "Understanding individual human mobility patterns," Nature, 453(7196), pp. 779-782, 2008.

[31] F. Calabrese, M. Diao, G. Di Lorenzo, J. Ferreira, and C. Ratti, "Understanding individual mobility patterns from urban sensing data: A mobile phone trace example," Transportation research part C: Emerging Technologies, 26, pp. 301-313, 2013.

[32] D. Taniar and J. Goh, "On mining movement pattern from mobile users." International Journal of Distributed Sensor Networks, 3(1), pp. 69-86, 2007.

[33] A. Chua, E. Marcheggiani, L. Servillo, and A. Moere, "Flowsampler: Visual analysis of urban flows in geolocated social media data," In International Conference on Social Informatics, pp. 5-17, 2014.

[34] J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M González, "Coupling human mobility and social ties," Journal of The Royal Society Interface, 12(105), 2015.

[35] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti, "Geo-located Twitter as proxy for global mobility patterns," Cartography and Geographic Information Science, 41(3), pp. 260-271, 2014.

[36] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services," International Conference on Web And Social Media (ICWSM), pp. 81-88, 2011.

[37] R. Harris, P. Sleight, and R. Webber, "Geodemographics, GIS and Neighbourhood Targeting," Journal of Direct, Data and Digital Marketing Practice, 8, pp. 364-368, 2007.

[38] P. A. Longley and M. Adnan, "Geo-temporal Twitter demographics," International Journal of Geographical Information Science, 30(2), pp. 369-389, 2016.

[39] D. Hristova, M. J. Williams, M. Musolesi, P. Panzarasa, and C. Mascolo, "Measuring urban social diversity using interconnected geo-social networks," In Proceedings of the 25th International Conference on World Wide Web, pp. 21-30, 2016.

[40] A. Llorente, M. Garcia-Herranz, M. Cebrian, and E. Moro, "Social media fingerprints of unemployment," PloS one 10(5), 2015.

[41] H. Abdelhaq, C. Sengstock, and M. Gertz, "Eventweet: Online localized event detection from twitter," Proceedings of the VLDB Endowment, 6(12), pp. 1326-1329, 2013.

[42] The one million tweet map, retrieved on April 8 2017 from http://onemilliontweetmap.com

[43] Tweepsmap, retrieved on April 8 2017 from https://tweepsmap.com

[44] Trendsmap Realtime Local Twitter Trends, retrieved on April 8 2017 from http://trendsmap.com

[45] MapD Tweetmap, retrieved on April 8 2017 from https://www.mapd.com/demos/tweetmap

[46] Mapping America: Every City, Every Block, retrieved on April 8 2017 from http://www.nytimes.com/projects/census/2010/explorer.html

[47] Social Explorer, retrieved on April 8 2017 from https://www.socialexplorer.com/explore/maps

[48] CitySense, retrieved on April 8 2017 from http://citysense.imis.athena-innovation.gr:8080/citysense/