Human Sensors: Case-study of Open-ended Community Sensing in Developing Regions

Kuldeep Yadav*, Dipanjan Chakraborty†, Sonia Soubam*, Naveen Prathapaneni†, Vikrant Nandakumar*, Vinayak Naik*, Nithya Rajamani†, L. Venkata Subramaniam†,Sameep Mehta*, Pradipta De‡
*Indraprastha Institute of Information Technology (IIIT), New Delhi, India
†IBM Research, India
‡SUNY Korea
{kuldeep, naik}@iiitd.ac.in {cdipanjan,lvsubram}@in.ibm.com

***Abstract*—With the growing number of cities and population, continuous monitoring of city's infrastructure and automated collection of day-to-day events (such as traffic jam) is essential and can help in improving life style of citizens. It is extremely costly and ineffective to install hardware sensors to sense these events in developing regions. Due to advent of smartphones, citizens can play role of sensors and actively participate in collection of the events which can be shared with others for information or can be used in decisions which affects city development.**

In this paper, we describe an architecture of crowdsensing testbed for capturing and processing events affecting citizens in cities in India. One of the design principle of our testbed is that it encourages users to do an open-ended sensing under five broad categories: Civic complaints, traffic, neighbourhood issues, emergency and others. As part of testbed, we allow events submissions from different submission modes i.e. mobile application, SMSes and web. Our mobile application exploits different sensing interfaces provided by today's smartphones to add contextual data with event reports such as images, audio, fine-grained location etc. Proposed testbed is used by university students across India to report event happening around them. Finally, we describe the data collected and uncover some of challenges and opportunities which may help future designs of crowdsensing based systems.

I. Introduction

Many developed countries have city-wide deployment of sensing infrastructure to collect data about day-to-day city events, the collected data is then analyzed online or offline to take a prompt action based on those events. For instance, USA have deployed traffic sensors across major highways to monitor the health of roads and to detect timely events such as traffic congestion, data collected from these sensors is also useful to make broad city development decisions. Such sensing infrastructure does not exist or have limited coverage in many countries due to lack of resources, cost, and bigger scale of deployment. Number of smartphones in the world are steadily growing and their are many sensors such as accelerometer, audio, GPS, camera etc which come with a smartphone. These sensors may be used for collecting rich and good quality data as compared to dedicated sensors deployment.

There has been several efforts to use smartphone sensors for variety of purposes such as to estimate pollution exposure [7], pothole detection [6] and traffic conditions [8]. All of these work expects people to participate, collect appropriate sensor data using their smartphones and contribute it for a common purpose, this process commonly is called as *crowdsensing*. Crowdsensing has emerged as a viable way of collecting rich data about day-to-day city problems without investing in specialized sensing infrastructure specifically in resource constrained environments. In Crowdsensing, the mobile phone and its sensors plays the role of *sensors* as well [1], aiding the task owner in the process. For e.g, in our investigation, the primary purpose of the task collector (or citizens) is to *sense the city*, i.e. any public events that a human senses in his/her surrounding, we are interested in receiving some information about it. Such information could be user's interpretation via text/voice updates, acoustic context, video, location, social vicinity, activities around the event.

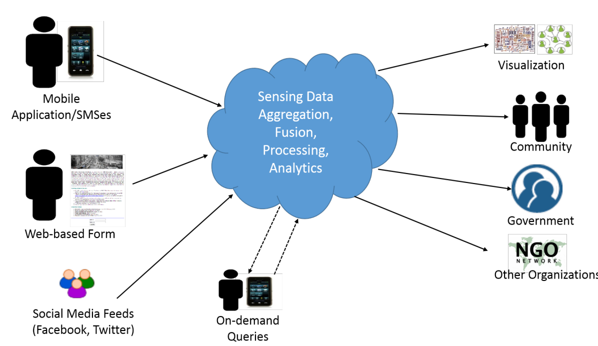

Figure 1: CrowdSensing and Multi-modal Data Fusion Testbed

We have built a testbed to collect city events using crowdsensing while offering different submission interfaces i.e Android-based mobile application, SMSes, and a web based tool as shown in Figure 1. Apart from this, our system automatically extracts events from given social media feeds. All the submitted events go to the Cloud which aggregates, pre-process these events, and then, find patterns from the data. Processed events data can be used to multiple purposes i.e. citizens information, city planning etc. Our mobile application provides an intuitive user interface to allow users to report events of a city falling under five broad categories: Civic complaints, traffic, neighbourhood issues, emergency and others. Mobile application make use of different sensors provided by smartphone and provides a choice to user to further, enrich the textual report with contextual data such as image and audio. For instance, if there is a traffic jam at some place B then user can submit a textual report "Traffic Jam at B due to break down of a truck", mobile application prompts user to take a picture showing the length of traffic jam, records the audio while fine-grained location is captured automatically using GPS. All the collected information are then sent to the Cloud. Unlike existing initiatives, proposed system provides a unified open-ended sensing interface to collect city's events with rich contextual data.

We have publicized our system among university students in India and asked them to submit city events happening around them. Here, we present brief analysis of submitted events to uncover some of challenges which we realized as part of our system deployment. The main aim of this paper is to present our preliminary effort to find the operating characteristics of smartphone-enabled sensing of our environment by the autonomous, human-driven "community of sensors". The future work would be focused on designing a sustainable and efficient community sensing infrastructure that maximizes quality of data, while being sensitive to the restrictions and limitations of this community of sensors.

II. System Details and Deployment

In our system, there are different modes to submit event reports i.e. (1) mobile application, (2) SMS (3) Web-based form and it automatically extracts event reports from social media feeds.

A. Android-based Mobile Application

We built an application for only Android OS due to two different reasons; (1) It provides rich support of APIs to capture contextual data (2) Android based mobile devices are getting increasingly popular in developing countries such as India. The Android application is designed in such a way that it provides a user friendly UI for the participants to report events with minimum efforts, snapshot of different screens of the application can be seen on Google play [1].

Whenever, a user wants to submit an event, she chooses a category which broadly describes the event viz. civic complaints, traffic, neighbourhood issues, emergency or others. After choosing an appropriate category of the event, application prompts user to enter more details about the event i.e. free form text describing the event, the location/landmark of the event, and some appropriate tags related to the event. To provide more contextual sensor information which can further assist the event report, the participant can also click the button *Click Image* which starts the camera of the phone and captures an image. The *Record Audio* button records a short audio clipping of 10 second duration to capture the sound in the vicinity of the event. On pressing the *Submit* button all the data including the text input, image, audio clip along with the GPS coordinates and cell information is uploaded using HTTP Post request to a server. Based on HTTP response, user gets a notification on their phone either acknowledging successful upload or an error message incase of a failure.

B. SMS and Web based Event Report Submission

In developing countries, there are many phones which have limited capability and may not have any programmable interface[2]. Also, there are significant number of phones which do not run Android OS. To extend the reach of our event report submission, we enabled participant to send report via SMS messages too. This option is suitable for non-programmable phones, non-supported smartphones, and users who do not prefer to use their data connection for sending reports.The following is a sample report: *Police asking 1000 rs bribe for approving passport for a friend, though all documents are perfect @ indore*

For non-Android smartphone or tablet users, we have also enabled web based event report submission. After one time registration and login, user can go to our website [3] and submit the even report similar to Android application. Our web-based form too allows rich data collection which can consist of text, audio and video inputs.

C. Social Media Feeds

Specifically in India, there are some initiatives started by government departments[4] and individuals[5] to crowdsense and disseminate information which may benefit citizens. Our system uses APIs provided by social media websites such as Facebook to extract event reports from them which may complement the data collection done as part of our testbed.

D. System Deployment

We have hosted our Android application on Google Play Store [6]. The challenge is publicized by directly reaching out to university students though emails and posters. There are also facebook and twitter pages to continuously engage the students. The system is currently running in real-world where participants are submitting reports and will be open

[1]http://goo.gl/dJK8Y

[2]http://www.newscientist.com/article/mg21528844.600-gps-workaround-helps-make-dumb-phones-smart.html

[3]http://kalpa.haifa.il.ibm.com:9080/indiaChallenge/

[4]https://www.facebook.com/pages/Delhi-Traffic-Police/117817371573308?fref=ts

[5]www.powercuts.in

[6]http://goo.gl/dJK8Y

till March 2013. Due to multiple event report submission methods, we capture different kind of information among which some of them need to be input by the user, while others get automatically captured and are sent when the event is submitted as shown in Table I.

Automatically Sent Details	**Manually Sent Details**
Time Stamp (S + W + MA + SM)	Event Type - (W + MA)
Lat,Lng,Cell Info (MA + SM)	Message/Text (S + W + MA + SM)
	Event Tags (W + MA)
	Textual Location (S + W + MA + SM)
	Image & Audio (W + MA + SM)

Table I: Automatic and manual information collected by different modes of our crowdsensing system i.e. SMS(S), Web-based form (W), Mobile Application (MA) and Social Media Feeds (SM)

III. Data Collection and Analysis

From our testbed, we are able to collect a total of 838 event reports among which 488 were submitted using web, 210 were submitted using SMSes and 140 events were submitted using Android application. A total of 435 users have registered so far in the system. While participants can submit text, audio and images, predominantly they have submitted text details of the events. Overall, we got 838 text reports i.e. all the submitted events had text, 182 of events had images and 5 also had audio clip. Most of text reports contained noisy data because intentional corruptions are very common in data uploaded from mobile devices [3]. This is due to the limited data entry options (keypad constraints on mobile devices) and due to the pressure of reducing communication latency (or cost in case of SMS) by keeping messages short yet intelligible.

We have used several pre-processing steps to extract meaningful information such as location, category etc from the reports which came from social media or SMS. For instance, many text reports do not have any delimiter which can be used to find the location names embedded into it. To find location name in such reports, we parse the text and use popular location suffixes such as 'nagar', 'chowk', 'cross' etc to estimate the location name. We have used a location dictionary to automatically learn location suffixes which can help in location extraction. Extracted location name could be a locality name or a city name, we used Google's geocoding API to convert the location names to approximate geo-coordinates but there may exist some location names which can not be geocoded by Google's geocoding service i.e. some lesser known locality names which does not exists on Google Maps [9]. For such names, we used a dictionary of 963 Indian cities and towns to translate the location names to their corresponding city names. Those city names were in turn fed to the Google's geocoding service to retrieve approximate geo-coordinates.

As part of analysis, we analyzed the distribution of different event categories across different states for finding major patterns in the submitted data. Across India, most of events (43%) submitted are about civil issues followed by traffic issues (22%), neighborhood issues (18%), emergency (7%) and others (10%). The three states from which we have received the maximum data are Uttrakhand (105), Delhi (95) and Tamilnadu (80). We have seen a large variance in distribution of various events categories across different states, for instance, maximum reports (54%) from Uttrakhand contains civil complaints; in Delhi, traffic events (42%) and from Tamilnadu, both civic and neighborhood issues (80%) were maximum.

(a) Traffic Issues in Delhi (b) Neighborhood Issues in Tamilnadu

Figure 2: Tag Clouds of some of major patterns found in data collection, Delhi residents were mostly concerned with traffic-related problems where as Tamilnadu had civic issues.

Further, we analyzed the event text for specific categories to find broad patterns in the data. Figure 2a shows the tag cloud of textual reports submitted for traffic related events in Delhi. Most of the event deals in reporting of high traffic, jam, congestion, water logging etc. Similarly, Figure 2b shows the tag cloud of report submitted for neighborhood issues in Tamilnadu and most of them deals with water logging, waster, drainage, unauthorized parking etc.

IV. Discussion

We have designed and implemented a community sensing testbed to sense various city's events or day-to-day problems with citizens's participation. The overarching goal of our effort is to explore the challenges in collecting balanced and reliable data by exploiting the unreliable, autonomous "community of sensors". Crowdsensing can be extremely useful for countries like India where due to large scale, it is very hard for automated sensors or city authorities to keep track of different events and taking prompt action. Different individuals have different biases in uploading data. As a result, the aggregate data is often biased towards a certain phenomenon (e.g. twitter has slowly become a news portal). Our goal is to design a ground up community-sensing architecture that keeps the community focused towards collecting urban environment data, while being sensitive towards personal preferences, restrictions. We have started with a simple community sensing testbed deployment and continuously expanding it to collect more and more data using human sensors which can be useful in long term. Here, we list some of our learning and future expansion plans in our testbed.

Participant Engagement : Similar to social networks, we have seen a power-law distribution where some of the participants are super contributors and a large percentage of participants report events intermittently. There are spatial trends too, total number of registered participants from Uttarakhand state is only 10 but several of them have been super contributors. But, Andhra Pradesh state with over 85 registered users has only 15 event submissions. In our testbed deployment, most of participants were college students, we are taking several measures for engaging users i.e. sharing reports with them, giving rewards to highest contributors [13] and even encouraging them to do an independent analysis to find patterns in sensed data.

Data Quality : From the collected event reports till now, we have seen that participant makes mistakes unintentionally while submitting data i.e. using noisy text in messages etc which complicates the analysis and requires lot of pre-processing before making it usable. Also, many times participants submits reports without giving sufficient details about an event i.e. without clicking a picture or recording an audio. Since, our testbed uses human in loop, we are working towards an automated mechanism to provide continuous feedback to every participant on her submitted reports so that in future, quality of reports can improve.

Data Validation : Crowdsensing based reports requires validation before they can be used for taking actions. We are using two different ways to validate participant's submitted events. (1) Correlation between submitted data : Among different data options, several integrity checks are made to validate the authenticity of an report. For instance, if human reported location in event report is same as automatic captured location. Also, if two or more independent participants report an event simultaneously, there is high degree of confidence in submitted report. (2) Challenge-Response Protocol: Our testbed allows firing remote sensing task to validate the submitted reports as described in [10], [11]. For instance, if system receives report of traffic jam at location A, it can fire a remote sensing task to a phone which is at location A to send more data (for instance, a picture or a audio clip) which can validate the submitted report.

Participant Overhead : Crowdsensing participants are volunteers with little or no incentives, system has to take special consideration to minimize costs (both monetary and energy) at participant's end. Submitting a single event transfers data of less than 0.5 MB in our android application which is reasonable. We are also working towards providing a cache functionality in mobile application where participant can record an event whenever it happens but upload it in bulk whenever there is free internet connection available i.e WiFi etc. Some sensing applications requires continuous sensing tasks, for instance road conditions monitoring requires continuous logging of accelerometer data. In such cases, mobile application should do optimizations such as in [12] to have minimal impact on battery and data connection costs.

Our data analysis in this paper is preliminary, as we collect more data using crowdsensing, we hope to observe more patterns and answer some of following questions (a) Can we extract trends in space or time dimension? (b) What modalities (text, voice, audio, photographs) should be used and their correlation with sensed events? (c) To what extent can we measure the real phenomenon from the sensed data? (d) Can we process and disseminate the collected data in real-time so that community can be benefit from it? (e) Can we automatically verify the authenticity of submitted reports? (f) To study characteristics and behavior of different participants in testbed?

Some of the above questions are long-term and will requires a good amount of data which we hope to collect in due course of time with the help of participants.

ACKNOWLEDGEMENTS

First author is supported by a PhD Fellowship from Microsoft Research, India. We would like to thank all the student participants who contributed in this crowdsensing effort and IBM Academic relations team who helped us in reaching out to universities.

REFERENCES

[1] Ganti R. K. et al. "Mobile Crowdsensing: Current State and Future Challenges", *Communications of ACM 2010*.

[2] Stewart O. et al. "Designing crowdsourcing community for the enterprise", *KDD Workshop on Human Computation 2009*.

[3] Subramaniam L V et al. "A survey of types of text noise and techniques to handle noisy text",*Workshop on Analytics for Noisy Unstructured Text Data 2009*.

[4] Goldman, Jeffrey, et al. "Participatory Sensing: A citizen-powered approach to illuminating the patterns that shape our world", Foresight & Governance Project, White Paper (2009).

[5] Yadav, Kuldeep et al. "Smsassassin : Crowdsourcing driven mobile-based system for sms spam filtering", *ACM HotMobile 2011*.

[6] Eriksson, Jakob, et al. "The pothole patrol: using a mobile sensor network for road surface monitoring", *ACM MobiSys 2008*.

[7] Min Mun et al. "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research", *ACM MobiSys 2009*.

[8] Mohan, Prashanth et al. "Nericell: rich monitoring of road and traffic conditions using mobile smartphones", *ACM SenSys'08*.

[9] Yadav, Kuldeep, et al. "Low Energy and Sufficiently Accurate Localization for Non-Smartphones", *IEEE MDM'12*.

[10] Ra, Moo-Ryong, et al."Medusa: A programming framework for crowd-sensing applications." ACM Mobisys 2012.

[11] Ravindranath, Lenin, et al. "Code in the air: simplifying sensing and coordination tasks on smartphones", ACM HotMobile 2012.

[12] Lu, Hong, et al. "The Jigsaw continuous sensing engine for mobile phone applications", ACM Sensys 2010.

[13] Samdaria, Navkar et al. "Paying in Kind for Crowdsourced Work in Developing Regions." Pervasive Computing 2012.