Building Apps with Open Data in India: An Experience

Mitesh Vasa
IBM Research Labs
Bangalore, India
mitesh.vasa@in.ibm.com

Srikanth Tamilselvam
IBM Research Labs
Bangalore, India
srikanth.tamilselvam@in.ibm.com

ABSTRACT
Open data is a well-established paradigm to make data available freely to everyone. The general belief is that open data leads to rapid pace in problem discovery, empowerment of citizens and greater collaborations. Opening up government data for free public access is a global trend, which India too followed in 2012. India is one of the early adopters, it has been ranked low in the last year’s Open Data Index. We participated in an open data app contest conducted by Government of India to come up with societal applications based on the datasets provided in data.gov.in portal. We would like to share our experiences and challenges during this contest and compare them with a similar internal contest that we participated in, where datasets were from U.S.

Categories and Subject Descriptors
E.m [Data]: Miscellaneous; H.5.2 [User Interfaces]: [Graphical user interfaces (GUI)]

General Terms
Design

Keywords
Open Government Data, India, Societal Application

1. INTRODUCTION
Open Data is an emerging phenomenon, gaining popularity not just with governments around the world, but also organizations like World Bank, United Nations etc. The 2013 Open Data Barometer [7], a joint initiative of WWW foundation and the Open Data Institute, listed 77 countries around the world that have published open government data. India’s official open data platform [2] got launched in October 2012 after the announcement of National Data Sharing and Accessibility Policy (NDSAP) and has been growing steadily since then. As of this writing, the portal claims to have datasets from around 59 different ministries such as Department of Finance, Home Affairs, Power, Petroleum, Agriculture etc., both at the national level as well as state/region level.

To leverage the open data, several governments conduct hackathon events regularly to engage citizens & professionals to participate in developing applications that benefit the city and its people. It also provides an avenue for entrepreneurs and small businesses to connect with venture capitalists and showcase their ideas. The open data movement has seen the rise of non-political organizations like ‘Code for America’ [1], whose goal is to connect application designers and developers with city governments and promote openness and participation. The academic community has recently organized several workshops around semantic cities with focus on open & linked data [12][13] (See the work on open data in corruption [16], healthcare [15], census, energy [11] etc.). The Government of India conducted an #OpenDataApps challenge in August last year in the same spirit. We built an app using data from the Department of Agriculture that helps users decide recipes based on real-time food prices. Although the verdict of the contest winners are not out yet, our experiences in obtaining & analyzing Indian agriculture produce data made us delve deeper into the nature of other datasets that are available on the portal. Through this paper, it is our endeavor to share the insights we gained with the community at large and point out the pros and cons of the current state of open data in India. We assess its quality on parameters not considered elsewhere such as the Open Data Index [3]. We also describe our experience in implementing the same app for an internal hackathon contest where the datasets available were different.

2. RASOI: FOOD FOR THE MASSES

2.1 Motivation
Recommender systems have become very popular in recent years, since they help users in precise, customized information retrieval despite the heavy data growth. Recipe recommender systems are also increasingly becoming popular like content-based and collaborative filtering approaches [19], hybrid-based [17], personalized-based [14] etc. There are even applications to find recipe similarity [18]. But we didn’t find any applications using open data that can provide insights into market price trends, allergies, nutrition etc. Rasoi is the Hindi translation for ‘cooking food in a kitchen’. We named our app “Rasoi: Food for the Masses”, to reflect its purpose - which is, to assist the common people, especially the home-makers, restaurant chefs, food caterers etc., in choosing food
ingredients, finding recipes, discovering food prices, grocery stores and so on from open data. The main motivation behind our app was to allow users to plan recipes/meals based on food prices of ingredients. From the societal angle, price fluctuations of fruits, vegetables and pulses due to inflation, demand-supply etc. cause a significant impact in the way people consume them. We wanted to capture this impact as well as provide an assist to users in terms of suggesting alternative ingredients and/or recipes that can tailor the expenditure patterns of middle-class households (which constitutes the majority of Indian socio-economic demography). In a developing nation like India, the average food inflation in the last 5 years (2008-12) hovered around 10.3%. In the year ending December 2013 itself, it was 13.68% [6]. The overall inflation index (and indeed, the food inflation index) of India is amongst the highest in the world, and the highest in Asia [5]. While the reasons are plenty, the key culprit for the high food inflation rate last year was onions, which in August had a 245% annual jump while other vegetables shot up by 77% [8]. Onion being a staple food ingredient in India, food caterers began to compromise on the quantities of onions and shallots used, whereas middle-class households began to cut back its consumption. Another interesting use-case of the data collected by the app was to provide government policy-makers a visibility into the impact of price rise on buying and consumption habits of people in near real-time, so that they can intervene proactively when needed.

subsectionDesign and Implementation We designed Rasoi as a web application with the look-and-feel of a mobile-app. The guiding principles for the design were to use as much open data as possible, simplify the process of finding recipes & use an intuitive user interface. We also kept the design generic so that it can be used for data from other countries/regions like China, Brazil, US etc. We used open source platforms & libraries like Java and JQuery and deployed it on Apache Tomcat, an open source web server. For those interested, the web-app is currently hosted on an Amazon EC2 instance for a limited period of time and can be accessed using a web browser 1. The app user is guided through a series of steps that help him/her select from a set of common, pre-defined

ingredients or even exclude ingredients that are undesired (either due to unavailability, high price, allergy, dislike etc.), followed by selection of a dish, a cuisine and a recipe. Users are expected to generally select ingredients that are currently available with them. Further, the app displays details of the selected recipe such as its ingredients, cooking time, the steps required to cook etc. along with an image of the final dish. The user can then select ingredients that s/he needs to purchase based on availability, price etc. and the app would display a map (if the user allowed the web browser to exchange geo-location information) of nearby grocery outlets within a range of 10 kms. The app then shows a table with all the store locations, store names, addresses, navigation details and the total cost of the selected food products. Although our intention is to allow users to select a store and place online orders, the current version of the deployed app does not have that feature. Figure 1 captures this process flow and Figure 2 shows a screenshot.

3. DATA

The #OpenDataApps challenge conducted by the Government of India had a key requirement: that the primary source of the dataset used for the app must be from its portal. Fortunately, the Ministry of Agriculture publish daily market prices of various commodities as reported from mandis (Hindi, for farmer’s market) across the country. The XML-based report contains the wholesale maximum, minimum and modal price of vegetables, fruits, nuts, pulses etc. from the various markets at district level. Since this dataset is made available as-is over the web and gets updated at no definite time interval, we had to build capability that would fetch the content of the web-page multiple times a day, parse the XML structure for the location & prices, and store the details in an in-memory map for efficient lookup by the app. Though the Open Data portal provides a data export functionality that generates the same dataset in JSON format, there is no mechanism to invoke it programmatically. Hence, we designed our app to make multiple HttpRequest calls in a day to fetch and parse the latest XML.

We soon realized that we don’t have all the data required for our app. Specifically, we didn’t have any data around cuisines, recipes, dishes etc. We decided to get a sample dataset from a popular Indian recipe website [10]. Since this website didn’t have any API or machine-readable file, we ended up creating our own formatted file and writing a parser for it. We also had to hard-code the prices of those ingredients that were not appearing in the online XML report. The bigger issue however, was that the prices of the various commodities were wholesale prices. We could not find any dataset for their retail prices, which are vastly different from wholesale prices due to transportation costs, preservation costs etc. The government runs a string of departmental stores across the country that provide the food products at reasonable prices for the public (as compared to the private sector retail outlets). But unfortunately, we could not find even their price list; so we ended up adding a random number to the wholesale price so as to ‘simulate’ the retail price. Next came the data requirement for the various store locations, their names, distances from the user’s current location etc. We needed an equivalent of Google Map and Google Distance Matrix API for India. Although the open data portal does provide a geographical map of India, it comes nowhere close to the data accuracy, coverage & requirements exhibited by Google’s APIs. Google provides a road network overlay with granular resolution, along with annotated data of stores in the vicinity.

Apart from the participation in the #OpenDataApps challenge, we also participated in an internal 48-hour hackathon event within our organization, where several APIs were provided for datasets exposed by US government/firms and the goal was to come up with innovative combinations of different APIs. We decided to apply the same idea to build an app that allows planning of meals, shopping for ingredients etc. However, our experience was vastly different. First, we didn’t have to generate or simulate data. There was US Naval cookbook data with a variety of recipes listed. We didn’t have to parse the dump of this data, as there were well-defined REST APIs with methods such as GetIngredients, GetRecipesByIngredient, GetCuisines, GetIngredientDishes etc. The API documentation was quite detailed in terms of query parameters, description, query data types, return data types etc. Not just that, the documentation provided sample code snippets for PHP, Ruby, Java etc. for several APIs using Unirest HTTP libraries. Programming using these APIs became extremely seamless, and we were able to focus more on the use-case, UI development etc.

Apart from the recipe API, we were also able to use Super-Market API [9] that has access to over a million grocery and home products at stores across the US such as Target, Walmart, Costco, Safeway etc. Their free API methods allowed operations such as searching for products by name, searching for stores by city, zip, state etc. For example, we could find names of all stores in the vicinity of a geo-point which keep products for ‘peanuts’. The API also gave us product meta-data like weight, description, image etc. Although the product price was part of the commercial API release and we could have subscribed for it, we decided not to go for it given the limited usage of the app post the 2-day hackathon. Instead, we implemented a simple randomizer function to generate prices. We then used Google Maps API to map the store addresses. Another advantage of using these three APIs was that all of them used JSON data format so we did not have to transform it from one form to another, like in the case of the XML mandi report.

4. CHALLENGES

While it is difficult to assess the state of Open Government Data in India, or for that matter any other nation, there have been several attempts in recent times. The Open Data Barometer of 2013 assessed 77 nations around the world on mainly three macro-level parameters: (a) readiness of government, businesses & society, (b) implementation of government initiatives and (c) their impact in bringing about political, economic and social change. It ranked UK the first, USA as second and India as 34th [4]. While this study was an initiative of Open Data Institute, the Open Knowledge Foundation launched a rather crowd-sourced approach for computing Open Data Index based on primarily the type of data released (i.e. transportation, budget, election, map, legislation, emission etc.) along key parameters such as whether the data is online, machine readable, up-to-date, free of charge, copyrights etc. As per this index, India ranked fairly poor - 63rd out of 70, whereas UK and USA maintained their positions. While these metrics evaluated the state of Open Data at a somewhat macro-level, we wanted to analyze the data more from a developer’s perspective. Based on
our experience, here are some specific as well as general observations & suggestions about the Open Government Data in India:

- **Data Organization & Coverage:**
  
  One common pattern we observed in datasets from several ministries is that the data can be organized in one file rather than split across hundreds of different files. For example, the Department of Power has 115 datasets for 'Power Supply Position', one for each month/year combo like April 2012, June 2009 etc. We found similar patterns in datasets such as 'Peak Demand and peak met', 'Progress Report of Village electrification', 'Monthly coal statement of thermal power stations' etc. In fact, all 411 datasets from Power Ministry followed this pattern (that is close to 7% of all datasets!). We browsed datasets from Water Resources Ministry and we found lot of redundant files. For example, the datasets for ‘Distribution of Shallow Tubewells’ are available as Village-wise, Block-wise, District-wise and State-wise separately, even though the village name, block name, district name & state details are present in the Village-wise dataset itself and hence one could easily infer the rest. The Ministry of Water Resources has over 1000 datasets, many of which fall in the same pattern such as ‘Distribution of Tanks/Ponds’, ‘Surface List Schemes’, ‘Distribution of dugwells’ and so on. The Ministry of Agriculture has around 1600+ datasets, one for each vegetable/fruit for every year in the last decade! The dataset supposed to give daily updates on UIDAI enrollments (Unique Identification Authority of India) is apparently not updated since last 3 months. Besides the data organization, we observed that there was very little information on some of the top issues facing India such as poverty, corruption etc.

- **Nature of Data:**
  
  Most of the datasets are provided through bulk access as downloadable flat files. There are no APIs defined that can be invoked programmatically. Rather, the onus is on the developer to parse the files and make sense out of the data. Although APIs are more beneficial when data is large and streaming, and incurs a cost in its development and support, it is always nicer to have APIs as they allow users to do mashups and retrieve data much more efficiently. For datasets that get updated daily at undefined time periods, like the Ministry of Agriculture’s commodity price list from various *mandis*, it makes even more sense to have APIs that can provide regular updates, otherwise it becomes a developer’s burden to find the delta at regular time periods.

- **Data Format & Schema:**
  
  Most of the datasets are available for download in either XML, CSV or XLS formats. As mentioned earlier, the portal provides a dataset export tool that can generate the data in JSON format, but there is no mechanism to invoke it programmatically. Hence, one needs to parse these formats or rely on external libraries to transform from XML to JSON and vice-versa, for example. In terms of schema, we noticed some datasets were described very well whereas for others, key metadata was missing making it tough to interpret the data.

For example, the Planning Commission’s dataset on ‘Percentage of AWC having toilet facilities’ was missing the exact year/period of data collection. We strongly recommend publishing a more complete meta-data, but also details on the data collection methodology - sample size, time spread over which it was collected, accuracy range etc.

- **Search:**

  Last, the search functionality of data portal needs to improve a bit. For example, for the dataset titled ‘All India and State Wise Area and Production of Vegetables’, if we search for the text ‘production of vegetables’, we are able to find the dataset but if someone searches for ‘vegetable production’, then the portal gives back zero results! The portal also doesn’t look for content inside the datasets, so a search for ‘graphite’ turns up zero results when it could have been datasets from the Ministry of Mines, for example. Besides improving the search capabilities, we request that perhaps a tag cloud based on keywords be made available to make the navigation easier & more visual.

5. **CONCLUSION & FUTURE WORK**

Through this paper, we make an attempt to analyze the current state of Open Government Data in India from an application developer’s point of view. We describe our experience in participating in the #OpenDataApp contest hosted by the Indian government, specifically in dealing with data required for our food recipe app, and relate it to the experience we had in building a similar app for a hackathon event meant for US market. While there are some challenges in obtaining, searching and fusing the open data in the Indian context, there is enough scope for improvement in these areas as well as the parameters measured by the Open Data Barometer and other similar studies. Given that the Open Data portal was launched very recently in 2012, it would be apt to let it mature and re-do this analysis a year or two later. At the same time, it would be interesting to read about other’s experiences in the Indian context as well as other countries.

6. **ACKNOWLEDGMENTS**

The authors would like to thank Biplav Srivastava & Vandana Srivastava for their help in guiding the design & development of the India societal app. Our sincere thanks to Shang Guo, Constantin Adam, Frederick Wu, John Rofrano & Sai Zeng for collaborating during the hackathon.

7. **REFERENCES**

Here we list the steps that one can follow to navigate through the application, along with corresponding screen-shots. Note that the application is available for a limited time at Amazon EC2 instance: http://54.200.19.247:8080/rasoi/index.html

**Step 0:** Launch screen (see Fig. 3)
This is the first screen that the user gets presented with.

**Step 1:** Select ingredients (see Fig. 4)
The user selects ingredients that are currently available (at home, for example) and also ingredients that they do not want in the dish they want to cook.

**Step 2:** Select cuisine/dish/recipe (see Fig. 5)
The user then optionally chooses a cuisine and the type of dish s/he is looking for. The recipe automatically gets filtered based on the user selection for cuisine and dish.

**Step 3:** View recipe (see Fig. 6)
The next step displays the recipe details like cooking time, complete list of ingredients, the steps to cook, a picture of the final product, nutrition chart (if available) etc.

**Step 4:** Select ingredients for purchase (see Fig. 7)
The ingredients required for the chosen recipe are then displayed along with their latest *mandi* prices of the day. A default value is shown for items not available in the XML report. User is expected to select at least one item.

**Step 5:** Visualize store locations (see Fig. 8)
Next, a Google map showing the names and locations of nearby retail grocery outlets is displayed. By default, a radius of 10 km. is considered. The browser may ask the user his/her permission to fetch the geo-location information.

**Step 6:** Show store details (see Fig. 9)
The last step displays additional store details like its name, address, navigation details, time to reach the store given the current traffic situation, aggregate pricing info etc.

---

**APPENDIX**

Here we list the steps that one can follow to navigate through the application, along with corresponding screen-shots. Note that the application is available for a limited time at Amazon EC2 instance: 

**Step 0:** Launch screen (see Fig. 3)
This is the first screen that the user gets presented with.

**Step 1:** Select ingredients (see Fig. 4)
The user selects ingredients that are currently available (at home, for example) and also ingredients that they do not want in the dish they want to cook.

**Step 2:** Select cuisine/dish/recipe (see Fig. 5)
The user then optionally chooses a cuisine and the type of dish s/he is looking for. The recipe automatically gets filtered based on the user selection for cuisine and dish.

**Step 3:** View recipe (see Fig. 6)
The next step displays the recipe details like cooking time, complete list of ingredients, the steps to cook, a picture of the final product, nutrition chart (if available) etc.

**Step 4:** Select ingredients for purchase (see Fig. 7)
The ingredients required for the chosen recipe are then displayed along with their latest *mandi* prices of the day. A default value is shown for items not available in the XML report. User is expected to select at least one item.

**Step 5:** Visualize store locations (see Fig. 8)
Next, a Google map showing the names and locations of nearby retail grocery outlets is displayed. By default, a radius of 10 km. is considered. The browser may ask the user his/her permission to fetch the geo-location information.

**Step 6:** Show store details (see Fig. 9)
The last step displays additional store details like its name, address, navigation details, time to reach the store given the current traffic situation, aggregate pricing info etc.
Recipe: Cabbage poriyal
South Indian Main Course
Cooking Time: 30 minutes
Ingredients:
- Cabbage, coconut, coriander, urad dal, haldi powder, red chillies, curry leaves, salt, oil

Steps:
1. Heat the oil, add the mustard seeds and urad dal and let the seeds crackle.
2. Add the green chilli, dry red chillies, curry leaves, and turmeric powder.
3. Add the coconut, coriander, and urad dal to the pan.
4. Place the cabbage in the pan, cover, and cook for 20 minutes.