My4Sight: A Human Computation Platform for Improving Flu Predictions

Vivek Bharath Akupatni

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Madhav V. Marathe, Chair
Naren Ramakrishnan
Jiangzhuo Chen
Keith Bisset

July 17, 2015
Blacksburg, Virginia

Keywords: human computation, human-in-the-loop, crowd sourcing, My4Sight, influenza forecasting

My4Sight: A Human Computation Platform for Improving Flu Predictions

Vivek Bharath Akupatni

(ABSTRACT)

While many human computation (human-in-the-loop) systems exist in the field of Artificial Intelligence (AI) to solve problems that can't be solved by computers alone, comparatively fewer platforms exist for collecting human knowledge, and evaluation of various techniques for harnessing human insights in improving forecasting models for infectious diseases, such as Influenza and Ebola.

In this thesis, we present the design and implementation of My4Sight, a human computation system developed to harness human insights and intelligence to improve forecasting models. This web-accessible system simplifies the collection of human insights through the careful design of the following two tasks: (i) asking users to rank system-generated forecasts in order of likelihood; and (ii) allowing users to improve upon an existing system-generated prediction. The structured output collected from querying human computers can then be used in building better forecasting models. My4Sight is designed to be a complete end-to-end analytical platform, and provides access to data collection features and statistical tools that are applied to the collected data. The results are communicated to the user, wherever applicable, in the form of visualizations for easier data comprehension. With My4Sight, this thesis makes a valuable contribution to the field of epidemiology by providing the necessary data and infrastructure platform to improve forecasts in real time by harnessing the wisdom of the crowd.

Acknowledgements

This work would not have been possible without the help and guidance of my committee members, and the colleagues at NDSSL laboratory.

First and foremost, I would like to express sincere thanks to my advisor, Dr. Madhav Marathe, for his invaluable inputs and guidance. He always had time for me despite his busy schedule and I learnt a lot from him. He was always supportive and encouraged my work.

I would like to thank Dr. Jiangzhuo Chen for his constant advice and making sure that I stay on the top of my schedule. I would also like to thank Dr. Naren Ramakrishnan and Dr. Keith Bisset for serving in my committee. I am grateful to Dr. Keith for his invaluable comments and suggestions during the design phase of the My4Sight system.

I would like to acknowledge the funding received for my research from DTRA CNIMS Contract HDTRA1-11-D-0016-0001, DTRA R&D V&V Grant HDTRA1-11-1-0016, NIH MIDAS Grant 2U01GM070694-09, and NSF NetSE Grant CNS- 1011769.

I am thankful to Mandy Wilson for her constant support and assistance during the whole process. Finally, I would like to thank my family and friends for providing me the necessary emotional support.

Dedication

To my mother, who has been the source of inspiration all through my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The word *epidemiology* originates from the Greek words *epic*, meaning "upon or among", *demos*, meaning "people", and *logos*, meaning "study or discourse" [41]. The word epidemiology, in its modern context, is defined as follows:

"Epidemiology is the study of the distribution and determinants of health-related states or events in specified populations, and the application of this study to the control of health problems." [66]

Although epidemiological study can take many forms [41], we are interested in forecasting the epidemic curve associated with an epidemic outbreak (for an example, see Figure 1.1). An epidemic curve displays either the daily or weekly number of infected cases observed during the outbreak period. The epidemic curve forecasting can be defined as follows: [51]

Input: Given an epidemic curve $y_1, y_2, y_3, \ldots, y_t$, where y_t denotes the daily or weekly infectious case count at time t.
Output: Predict $y_{t+1}, y_{t+2}, \ldots, y_n$, where n represents the epidemic outbreak period.

A number of mathematical and computational models exist for forecasting the dynamics of infectious diseases. Among them, the classic compartmental (population) models are the most widely used. These forecasting models split the population of interest into different compartments, and allow people to move between the compartments, such as suspected to infected, or infected to dead (or recovered). Although these models are parsimonious in nature and are relatively easy to set up, they make simplistic assumptions that people within a compartment have uniform mixing, and the models don't account for dynamic adaptation of human behavior in response to the developing epidemic.

At the other end of the spectrum of compartmental models are agent-based models (ABM), where the population is represented as a network, and each node corresponds to an individual, a software agent in a virtual world. The agents' movements and contact patterns in the virtual world are modeled by the edges in the network. ABMs are simulated

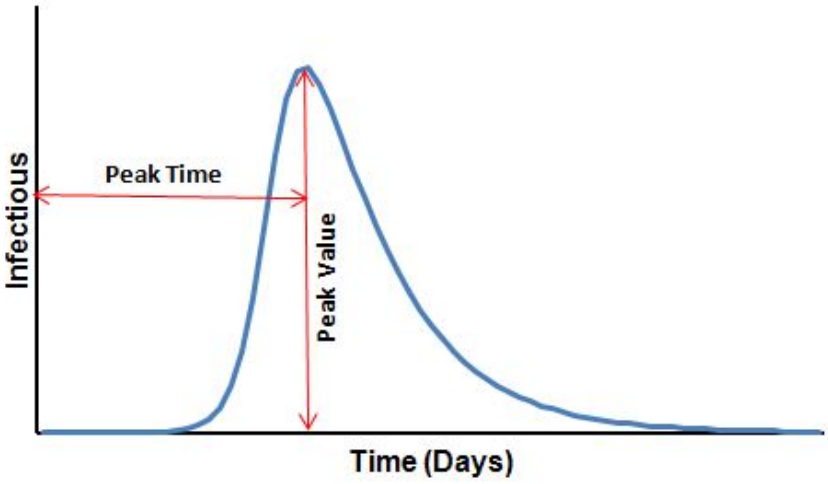


Figure 1.1: Epidemic Curve

over a certain disease model, such as Susceptible-Infectious-Recovered (SIR) or Susceptible-Exposed-Infectious-Recovered (SEIR). Each agent is assigned a certain disease state before the beginning of the simulation, and, additionally, agents are assigned rules which mimic human behavior, such as fear, bias, etc, seen in real-time epidemic scenarios. During the course of the simulation over several replicates, the model keeps track of agents, and updates their states as they move around in the virtual world. ABMs provide the opportunity to explore how individual behavior affects epidemic dynamics, and allows for the evaluation of the effect of various intervention scenarios, and thus plays an important role in determining public policy decisions. However, deployment of ABM models requires realistic contact networks with fine-grained geographic and demographic information which may be unavailable for a particular region. Another challenge involves coming up with the right set of rules to describe human behavior, and this involves common sense knowledge and guesswork because of our limited ability to understand human behavior [52, 34, 50]. For an in-depth review of different forecasting models and their limitations, see [63]

Apart from the models based on traditional surveillance data sources, new models have been developed using novel data sources such as climate data [68, 71, 67], using digital data sources such as Google search keywords [53], Twitter [60, 55], HealthMap, and table reservation data available through OpenTable APIs to forecast Influenza-like Illness (ILI) case counts. Additionally, systems [57, 49, 47] have been developed which make forecasts by combining multiple data streams. However, few studies [43, 44] indicate that digital data sources need not always provide good signals for Influenza trends.

All forecasting approaches discussed above make certain assumptions regarding the infectious disease, such as disease transmission, effect of disease control measures, etc., and exhibit limitations in modeling the real epidemic outbreak given the challenges in obtaining reliable

surveillance case data. Thus, uncertainties and inaccuracies are a part of these models and affected by countless variables. In these uncertain scenarios, how does an epidemiologist confirm the strength and weakness of his models and determine if there is a vector for improvement? Work done by Polgreen et al. [65] indicates that microbiologists, pharmacists, field experts, physicians, and nurses can provide useful insights regarding future epidemic activity. We believe that by crowdsourcing, and identifying the right people in the crowd, the uncertainty of the models can be greatly reduced, and the model's predictive accuracy greatly improved. To achieve this vision, we developed the My4Sight platform.

1.1 Motivation

Flu season is an annual event in the United States due to the prevalence of the influenza virus. The Centers for Disease Control and Prevention (CDC) report that the flu vaccine is only 60% effective in all the cases. Recent estimates put the annual influenza cost to the US economy at $71- $167 billion [6] due to hospitalizations, missed work days, and lives lost. It is estimated that roughly 5 to 20 percent of the US population contracts influenza every season, resulting in approximately 36,000 deaths [38]. Reliable and timely predictions of flu dynamics could help public health officials in making informed decisions concerning allocation of resources. Earlier prediction of flu forecasts can enable policy makers to choose the appropriate control and preventive measures so that the impact of the disease is reduced.

However, reliable forecasting of influenza epidemics is challenging for the following reasons: (i) each flu season exhibits variations in timing and intensity with respect to the previous year's outbreaks; (ii) non-availability of reliable surveillance data; and (iii) the incidence rate depends on climatic, behavioral (contact patterns, age groups) and biological factors (immunity etc) [63], which are difficult to account for. This situation is further complicated when a novel strain of the virus is introduced, as was the case during the 2009 flu season.

1.2 Contributions

Here, we present My4Sight, a system to harness human computation to improve epidemiological forecasting models. Our main contributions could be summarized as follows:

1. **Human Computational Tasks:**
   Our primary contribution includes the design and development of two tasks, i.e. ranking and individualized forecasting, to gather human insights regarding evolving epidemics. These human judgments can be used to inform and improve epidemiological forecasting models.

2. **Analytical Platform:**
   The back end of My4Sight is designed as an analytical platform, providing access to flexible queries, data collection features, and machine-learning algorithms.

3. **Generic Extensible Platform:**
   Epidemic forecasts are based on several factors which include the type of disease, geographical region, data source, and forecasting model; therefore, the platform is designed to handle a diverse set of epidemiological forecasting models, surveillance data sources, multiple diseases, and geographical regions. The platform also stores historical surveillance data to provide useful insights and trends.

4. **RESTful Web Service APIs:**
   The entire platform is implemented using several independent components which interact with each other using well-designed APIs. The front end and back end are completely decoupled, and data only flows through RESTful APIs. As a result, epidemiological data and machine learning algorithms are available to other researchers and third-party sources as web services.

Chapter 2

Background and Related Work

2.1 Human Computation Systems

The term "human computation" in its modern context was coined by Luis von Ahn in his doctoral dissertation(2005) [72]. He broadly describes human computation as a method for tackling difficult problems which are either computationally intensive or unsolvable by computers alone by harnessing the combined intelligence of humans and computers. His seminal work includes the creation of CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) [36], which are used in most of the popular websites to distinguish a human user from an automatic bot, and Games with a purpose (GWAPs), enjoyable online games where humans help generate useful data (e.g., annotating images, videos or music labels) or contribute in small ways in solving a larger problem (e.g., protein-folding tasks [11], graph coloring [56]). Although the idea of using humans to solve large computational problems is well documented throughout history [54], the growth of the Internet has made it possible to harness human intelligence on a larger scale.

A human computation system can be defined as follows [59] :

"An intelligent system that organizes humans to carry out the process of computation – whether it be performing the basic operations, taking charge of the control process itself, or even synthesizing the program itself"

In order to achieve system objectives, a human computation system (HCS) needs to recruit humans to complete its tasks. One of the popular ways to recruit human beings is through crowdsourcing. Howe, Jeff (2006) [12] defines crowdsourcing as follows:

"Simply defined, crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole

individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers."

The majority of the systems [69, 64, 58] use crowdsourcing as a way to recruit workers, although systems can use a closed set of workers to distribute tasks.

Three important aspects of any human computation system [59] are shown in Figure 2.1. In the following section, we will describe the role played by each of them briefly.

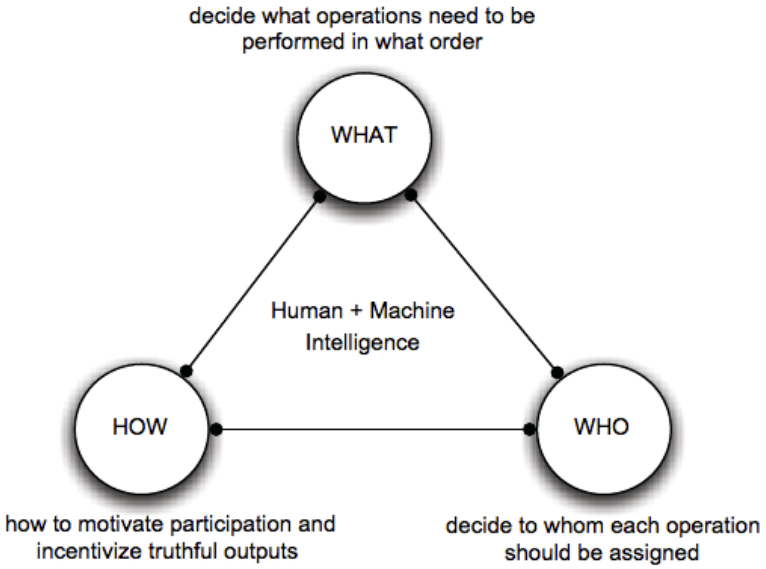


Figure 2.1: Three central aspects of human computation systems

- **What Aspect**: Addresses the central issue of how to design an algorithm which solves a given computation problem by using humans in a most efficient and accurate manner. This aspect deals with recasting a complex computational problem into a well-defined set of operations that can be carried out by workers. For example, Soylent [45], a crowdsourced word processing interface, provides the proofreading service in the following manner: (i) In the **Find** stage, workers identify sentences/paragraphs that need to be reworked; (ii) in the **Fix** stage, workers fix the regions identified in find stage; (iii) and in the **verify** stage, workers vote on the fixes provided. Additionally, the system should be able to handle the complex and noisy outputs from multiple human computers, as humans can make mistakes and exhibit variability in judgment and opinions.

- **Who Aspect**: Not all human computation tasks are equal, and a few of them are knowledge-intensive, requiring domain expertise. Task routing addresses the problem of assigning tasks to workers in a way that ensures that each worker has the necessary

skills to complete his task. There are two broad approaches for task routing. (i) In the **Push** model, the system plays an active role in task routing, and determines which workers can complete a given task. To accomplish task routing in the push model, the system needs to take into account worker characteristics, like availability, expertise, interest, and competence. (ii) In the **Pull** model, the system plays a passive role in task routing, merely providing the necessary platform/tools for workers to assign tasks for themselves. The workers use search interface functionality provided by the system to find and complete their tasks. The tools provided for browsing, searching and visualization of tasks play an important role in pull models. In human computation games like FoldIt, the task routing becomes even more complicated as system needs to strike the balance between difficulty and fun when assigning a task to worker, and workers expect tasks to be assigned to them even when there are no active tasks available within the system.

- **How Aspect**: The design of questions that are posed to human computers plays an important role in eliciting truthful and accurate responses from the crowd. The system should be easy to learn and use; users can also be motivated to revisit the system and provide additional units of computational work if they are offered something that has significant value to them. Human computers can differ greatly in terms of motivation; beyond monetary incentives, factors that can encourage workers to contribute to the system include:
  - Interest in accessing resources that they consider important to them (e.g., MedWatcher[28])
  - The quest for fun or entertainment in their leisure time (e.g., GWAPs, Games with a Purpose like FoldIt)
  - Willingness to participate and contribute to community welfare, digital volunteering during disaster and mass emergencies, or advancing the state of science (e.g., Ushahidi[32], FluNearYou[10], citizen science projects, and other crowdsourced crisis informatics tools).

2.1.1 Popular Human Computation Systems

- **FoldIt**

  FoldIt [11] is an online puzzle game aimed at solving protein prediction problems by harnessing the superior pattern recognition skills of humans. Protein folding is considered to be one of the hardest problems in biology today, and human intervention can help scientists find better treatment for diseases. In this online puzzle, users are asked to reshape improperly folded protein sequences to the state which has the lowest energy. The configuration with the lowest energy score gets the highest number of points. The system uses gaming concepts to encourage users' participation and contribution.

- **Wikipedia**

  Wikipedia [39] has become the go-to place for all the standard information on the web, and acts as an online encyclopedia which can be edited by anyone, anytime. The main idea is to harness the combined power and skills of all the users who browse the web to create the ultimate online reference. Wikipedia is the most popular of the wikis, and currently has around 4.9 million [42] English articles. Wikipedia has a well-functioning online community (i.e., people who read, edit, moderate, and administer content), which helps prevent pages from being vandalized. The system provides tools that make it easier for the community to report and correct page disruption, neutrality and vandalism.

- **Duolingo**

  Duolingo [3] is a recent entrant in the list of popular crowdsourcing applications with over 100 million registered users around the world [4]. The aim of this platform is twofold: (i) to translate the content on the web into multiple languages, and (ii) to help users learn new languages. From the user's point of view, Duolingo is a free language learning tool that provides personalized content and uses data-driven methods to teach a new language. From the system's standpoint, the user's answers to exercises are used in to translate the content from one language to another. The human computation book [59] estimates that the cost of translating the English Wikipedia to the Spanish version would be $50,000,000 (assuming Amazon Mechanical Turk workers are recruited for this task at a rate of 5 cents per word). In order to be self-sustaining, the system generates revenue by translating articles for its clients.

2.2 Related Work

To the best of our knowledge, Polgreen et al.[65] was the first to demonstrate that experts' opinions can be aggregated to forecast infectious disease activity. The pilot study was conducted for the state of Iowa for the 2004-2005 influenza season, and experts' subjective and privileged information was aggregated using a prediction market. The experts were asked to predict the influenza activity level in Iowa, and had to choose from five different activity levels (widespread, regional, local, sporadic, and no activity) used by the Centers for Disease Control and Prevention's (CDC) color-coded system. The prediction market could successfully predict the epidemic activity level at least 2 weeks in advance. Also, the study found that, in general, active users outperformed inactive users in predicting the influenza activity, suggesting that active users were better informed.

Epicast [2] : Our opinion is that this tool developed by Delphi research group [1] at CMU closely matches our My4Sight system. At this time, Epicast asks users to make individualized predictions for the CDC HHS regions. The system website mentions that the research group's intention is to aggregate predictions from users, and also a subset of users, and compare them

against forecasts made using other techniques. Also, analytics support doesn't seem to exist in the current version of the system. The system dashboard includes a leaderboard to drive competition among users to make better flu predictions.

2.3 Prior Work on My4Sight

My current work builds upon the previous work done on the My4Sight application. The previous version contains a ranking widget which allowed users to rank system predictions, as shown in Figure 2.2. However, this ranking widget was not designed to support more than 3 predictions at a time. The earlier system didn't have the ability to support multiple diseases, models and data sources, thereby limiting its adoption and usage. The backend data model didn't support flexible queries, lacked features to model user responses, and user activities on the system were not tracked. The following chapters explain how these issues are addressed in the newer version of the system.

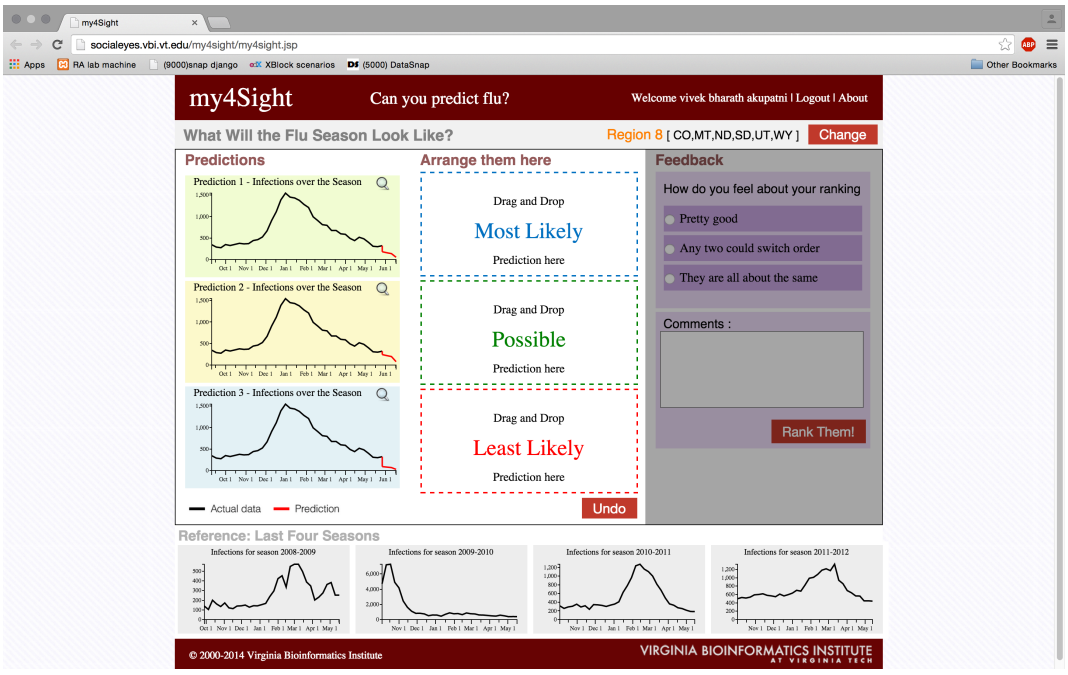


Figure 2.2: Previous version of my4sight System, http://socialeyes.vbi.vt.edu/my4sight/my4sight.jsp, Used with permission of Dr. Madhav Marathe, 2015.

Chapter 3

My4Sight Human Computation Tasks

A careful design of computational tasks is necessary for any human computation system to be effective. The human computation book [59] deals with the design of tasks in a comprehensive manner and provides guidelines for future human computation systems. We will discuss a few of them which are applicable in the context of My4Sight. Any operation that needs to be performed by human users needs to be articulated well with a clear set of instructions to reduce the scope for misinterpretation. Also, usability plays an important role in human computation systems [59]. The system should be easy to learn, and provide intuitive functionalities so that it encourages users to return to the system to complete more tasks. Earlier studies [48, 73] indicate that tasks which cannot be performed in a reasonable amount of time can discourage users from using the system.

How can we harness human intelligence to solve the system's objective of making a more accurate weekly **I**nfluenza-**l**ike **i**llness (**ILI**) case count? One possible way to do this is by asking open-ended questions to the user directly [31]. Consider the following question:

- What do you think the weekly ILI case count will be in week 10 for HHS region 1?

The above question could also be framed in ILI weekly percentages. However, this kind of open-ended question is cognitively overwhelming, and may result in less accurate responses from the user. The system [31] tries to decompose the above task into a simpler question by providing some predefined sets of options for users to choose from. The above question is recast in the following way:

- What do you think the weekly ILI case count will be in week 10 for HHS region 1?
    1. >20000 (more than 1%)
    2. >40000 (more than 2%)
    3. >200,000 (more than 10%)

The studies [11, 61] found that providing relevant information along with the question, such as partially solved questions or asking users to iterate (improve) over existing solutions, seems to provide more accurate results from the workers. The Epicast [2] tool from Delphi's [1] forecasting group asks users to predict the weekly ILI case count by using a drag and drop widget. Inspired by the above works, My4Sight tries to involve humans by showing them forecasts prepared by either data-driven or causal models (these forecasts are considered partial solutions) and defines the following two tasks:

1. Ranking: This task asks user to rank system-generated forecasts in order of likelihood.
2. Making individualized predictions: Users are allowed to improve upon an existing system-generated prediction by using a custom drag and drop html widget, similar to one used by Epicast.

The task representation employed in My4Sight helps to convey the temporal relationship inherent in disease epidemics, which is useful in getting better responses from the crowd. The user responses from the two tasks above are stored in a relational format in databases, and this data can later be fed into machine learning algorithms to improve the backend models. The next sections in this chapter describe these tasks in detail, and strategies for motivating users is considered at the end of this chapter.

3.1 Ranking Task

Task Description: "Place System Predictions in Order by Dragging and Dropping (Most likely prediction first, and least likely prediction last)"

Input: Users are provided an ordered list of predictions $P = [p_1, p_2, \ldots, p_i]$ where $1 \leq i \leq K$. K represents the total number of system-generated predictions for the ranking task, and p_i represents individual forecasts.

Output: A permutation which is the binary bijection from P to itself. We will denote $\pi(p_i)$ as the rank given by a user to prediction p_i. The result can also be treated as set of tuples $O = \{(p_i, \pi(p_i)) \mid p_i \in P\}$. The user's response produces output O, which is easy for computational consumption by data analytical methods.

In this task, the users can rank the forecasts generated by the system from most likely to least likely to occur. For this purpose, the system employs a ranking widget which provides a positive user experience by allowing users to drag and drop the forecast plots. Figure 3.1 shows the previous version of My4Sight's ranking widget; this widget had following drawbacks:

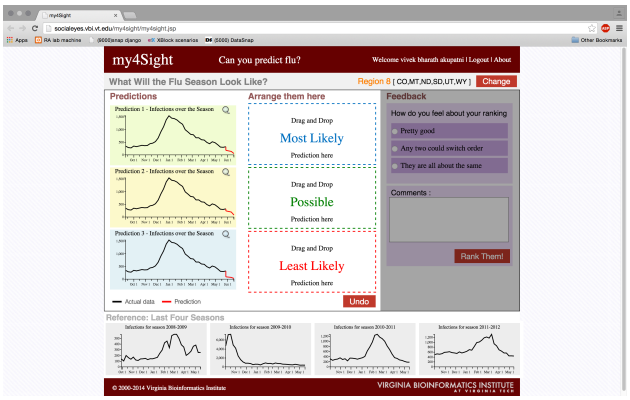


Figure 3.1: Earlier Ranking Widget, http://socialeyes.vbi.vt.edu/my4sight/my4sight.jsp, Used with permission of Dr. Madhav Marathe, 2015.

1. Inflexibility to support more than 3 prediction plots for ranking
2. Difficulty in embedding My4Sight inside html iframes (to create mashup web applications) due to the fixed layout design
3. Memoryless – The ranking widget state is not updated when the user completes a task. This is a poorly designed information flow architecture because the users don't see their earlier ranking responses, and it increases the cognitive load since they'll need to explicitly remember previous results.

To overcome the above mentioned difficulties, a newer version of the ranking widget was developed. Figure 3.2 shows the prediction curves as seen by the user before ranking, and Figure 3.3 shows the state of the ranking widget after the user saves the ranking task. When the user logs into the system at a later stage, his/her previous ranking responses are communicated in a similar fashion as illustrated in Figure 3.3, and Figure 3.4 shows an example of ranking tasks involving 4 system-generated predictions.

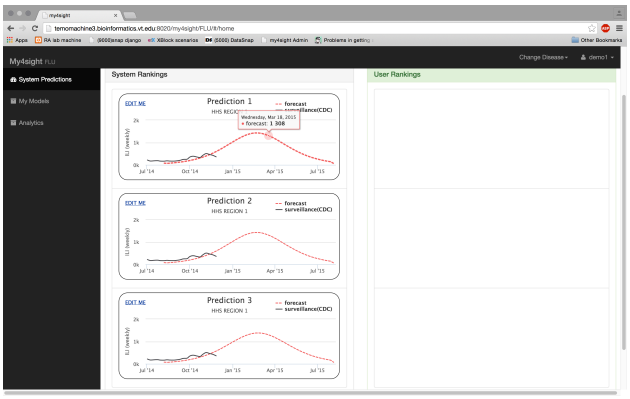


Figure 3.2: Ranking task before completion. System Rankings on the left panel shows the forecasts generated by the system for the user to rank. The user can complete the task by dragging the forecast curves from the system ranking panel on the left to the user ranking panel on the right, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

Although users can update their ranking results by canceling earlier answers, the backend is designed to preserve all the answers. In order to prevent any bias while ranking prediction curves, the users can only see the aggregate opinion of the crowd once they complete the ranking task. Figure 3.5 shows the vote distribution for one of the ranking tasks completed by the user.

3.2 Individualized forecasts

Task Description: "Make improvements/modifications to the system prediction by directly manipulating the prediction curve"

Input: $O = \{o_1, o_2, \ldots, o_T\}$ is the observed **I**nfluenza-**L**ike **I**llness (ILI) case count for a region under consideration, and T is the time period in which the ILI case count is known. Let $P = \{p_1, p_2, \ldots, p_T, p_{T+1}, p_{T+2}, \ldots, p_n\}$ represent the forecasted epicurve where $n > T$.

Constraints: When making individualized predictions, the user can modify all weekly case counts p_t where $t > T$. The past ILI case counts $p_t \; \forall t \leq T$ cannot be dragged (or modified) by the user.

Output: The output from the above-mentioned task results in the generation of a new forecasted time series $P' = \{p_1, p_2, \ldots, p_T, p'_{T+1}, p'_{T+2}, \ldots, p'_n\}$. It should be noted that modification of a single case count is also treated as a new forecasting model. The resulting time series P' captures the knowledge (or intuition) of the user about the ongoing influenza season, and this gets stored in the backend database along with its associated user login credentials. The user model P' can then be used to revise backend models to make better

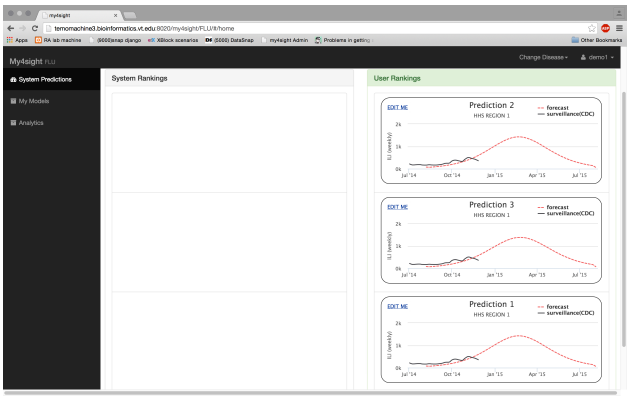


Figure 3.3: Ranking widget After Saving, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

predictions if the user has made consistently good predictions in the past.

Unlike the previous ranking task, this task provides direct access to the forecasted curve and allows users to express their experiential knowledge. For this task, a new html widget is added to the front end which allows users to make individualized predictions by manipulating an existing prediction made by the system. This allows subject matter experts to submit their own beliefs about how the season is headed and to express (or modify) their wealth of information regarding epidemic forecasting measures like peak value (or time), first-take off value (or time), intensity duration, velocity of the epidemic, and the start time of flu season.

Figure 3.6 shows one of the forecast plots made by the system before user improvements/-modifications, and Figure 3.7 shows the new forecasting model created by the user after modifying the system-generated prediction. The "My Models'" tab on the main dashboard shows the list of individualized models created by the user. For an example, see Figure 3.8. To facilitate identification of the different forecasting models created by the user, we ask users to enter a unique name for each new forecasting model they create. Also, the individualized ranking widget allows users to revert their changes by providing "undo" and "cancel" options.

3.3 Incentive Structure

Strategies for motivating humans to participate and complete computational tasks can vary. Nonetheless, incentives play an important role in getting better structured and truthful responses from the crowd, and impacts their level of efficiency and participation.

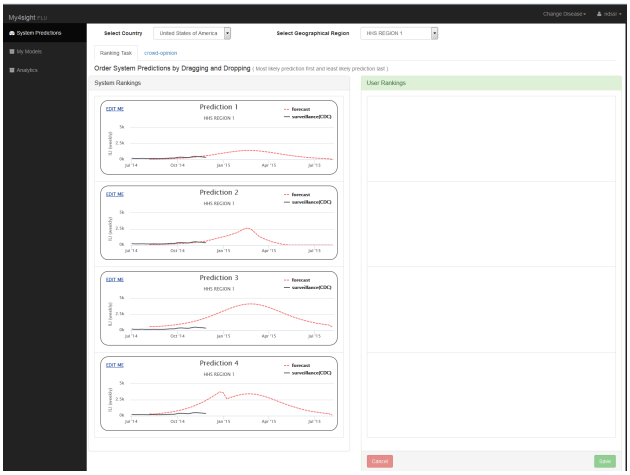


Figure 3.4: Ranking widget for 4 system-generated predictions, http://epics.vbi.vt.edu/my4sight/,
Used with permission of Dr. Madhav Marathe, 2015.

3.3.1 Extrinsic Motivation

Two kinds of external incentives could be supported by the system, and we envision that these will be explored in depth and implemented in the future. These two external incentives are described below:

1. Monetary Payment: Simple ranking tasks can be processed by workers from micro-incentive crowdsourcing platforms like Amazon Mechanical Turk (MTurk) [8] in exchange for a small monetary payment. This can be accomplished by launching an external Human Intelligence Task (HIT) [7] in MTurk which can render the My4Sight web application inside a worker's web browser. The external HIT task provides the flexibility and control over what tasks are displayed to the worker and how the results are collected at runtime. The newer version of My4Sight's front end is implemented from the ground up using a responsive web design [30]. The responsive web design enables websites to automatically adapt the content of the site to screen sizes of various web-enabled devices, including iframes. As a result, the application adapts perfectly well as the browser window size is changed. This feature, along with modularized user authentication on the back end, allows My4Sight to be easily extended to work with MTurk.

2. Leaderboard : Users can be awarded virtual points based on the quality of their forecasting model, and the dashboard could be modified to display cumulative points to drive competition between users.

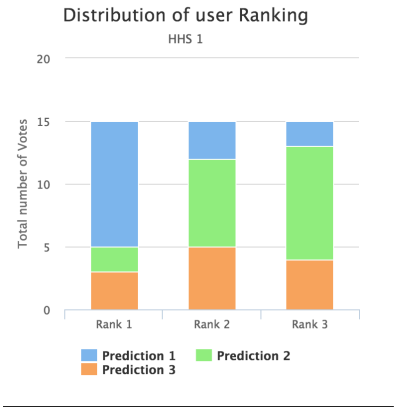


Figure 3.5: Crowd Opinion for the System Predictions (Ranking Task) indicates how users ranked system-generated predictions, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

3.3.2 Intrinsic Motivation

From an expert user's point of view, the computational problem that My4Sight is designed to solve has significant value, and thus, such users are intrinsically motivated to provide their insights. When the true value for the forecasted event is available, the user can re-assess/reevaluate their knowledge and respond accordingly. The system provides access to data collection features and machine learning algorithms that are applied to the collected data. The results are communicated to the user, wherever applicable, in the form of visualizations for easier data comprehension.

3.4 Privacy Concerns

The system requires users to log in before they can complete new tasks or view earlier results. This is done to minimize spam because of bad actors or bots. We believe that most responders using the system would be genuine and willing to provide their individual opinions about the evolving flu season. However, in order to prevent users from replicating each other's responses, the system ensures that rankings and individual forecasts predicted by a user are not visible to the crowd. Although users cannot see others' predictions, the system does provide aggregate crowd opinions, either in the form of visualizations or summary tables.

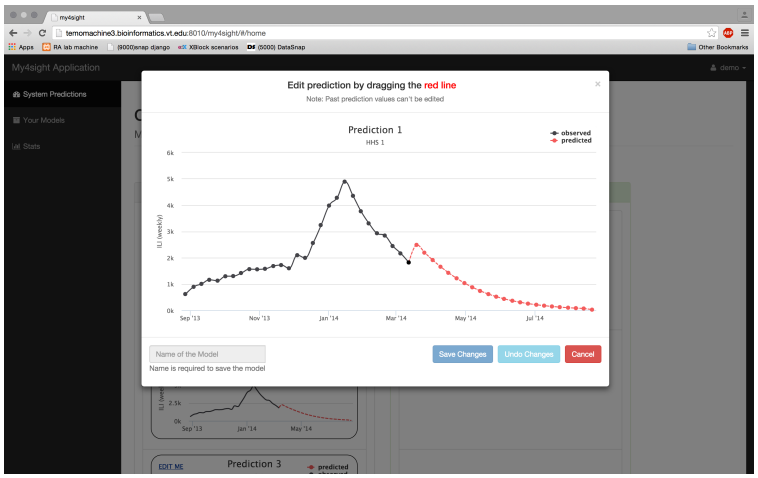


Figure 3.6: UI Widget which allows user to drag and drop the forecasted curve. The widget provides options for users to save, undo, or cancel their work, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

Figure 3.7: Prediction Curve After User Modifications, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

Figure 3.8: List of User Models, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

Chapter 4

My4Sight System Architecture

This chapter will describe the system architecture of My4Sight. The architecture is designed to provide a flexible, scalable, and web-based analytical platform to support human computation tasks on epidemic forecasts.

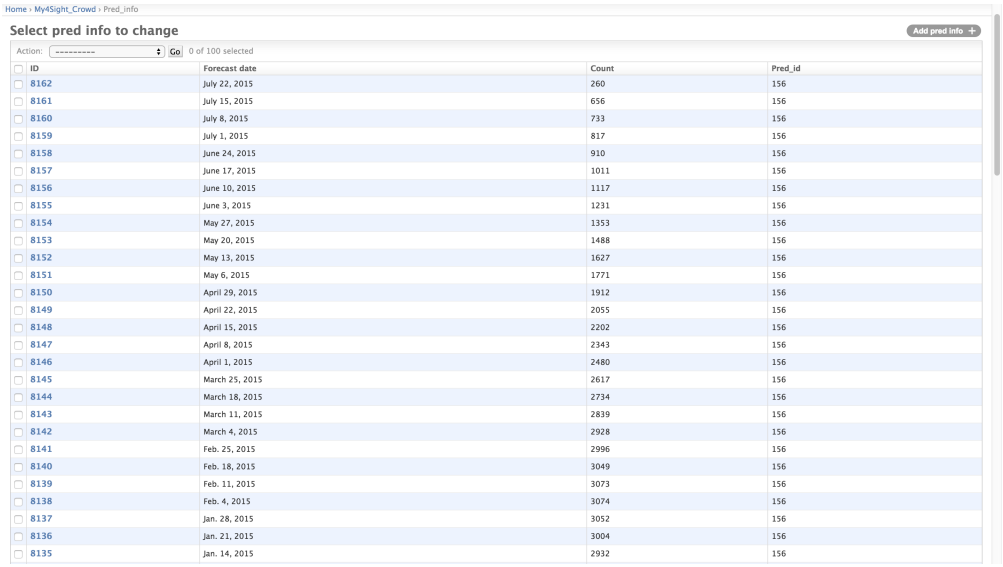
4.1 Design

The design and implementation of the My4Sight back end is greatly influenced by the following factors. The key features of My4Sight include:

- Multiple Dimensions for forecasting models: The epidemiological models are functions of several input parameters, and the system is not restricted to a particular disease or model. The system accounts for diversity along the following epidemic dimensions:
  - Infectious diseases
  - Geographical regions
  - Forecasting models
  - Uncertainty bounds for models
  - Seasons (time periods)
  - Surveillance data sources
  - Quality metrics for evaluating forecasting methods
- Data Ingestion:
  - Automation: In order to streamline the process of populating necessary surveillance and forecast data, a set of python scripts were developed. These scripts

are intended to integrate My4Sight with other forecasting pipelines supported by NDSSL (**N**etwork **D**ynamics and **S**imulation **S**cience **L**aboratory).

– Functional User Interface (UI): The system assumes that end users who configure the human computation tasks as described in Chapter 3 should not have to be tech savvy to develop scripts for populating data from different surveillance and forecasting data sources. To make the system more approachable, the My4Sight back end includes a functional UI which can be used for data population. This UI abstracts many of the complex details by providing a simple web form for entering and modifying data dynamically. For an example, see Figure 4.1.



ID	Forecast date	Count	Pred_id
8162	July 22, 2015	260	156
8161	July 15, 2015	656	156
8160	July 8, 2015	733	156
8159	July 1, 2015	817	156
8158	June 24, 2015	910	156
8157	June 17, 2015	1011	156
8156	June 10, 2015	1117	156
8155	June 3, 2015	1231	156
8154	May 27, 2015	1353	156
8153	May 20, 2015	1488	156
8152	May 13, 2015	1627	156
8151	May 6, 2015	1771	156
8150	April 29, 2015	1912	156
8149	April 22, 2015	2055	156
8148	April 15, 2015	2202	156
8147	April 8, 2015	2343	156
8146	April 1, 2015	2480	156
8145	March 25, 2015	2617	156
8144	March 18, 2015	2734	156
8143	March 11, 2015	2839	156
8142	March 4, 2015	2928	156
8141	Feb. 25, 2015	2996	156
8140	Feb. 18, 2015	3049	156
8139	Feb. 11, 2015	3073	156
8138	Feb. 4, 2015	3074	156
8137	Jan. 28, 2015	3052	156
8136	Jan. 21, 2015	3004	156
8135	Jan. 14, 2015	2932	156

Figure 4.1: UI for entering data into the system. In this example, the UI contains a web form to modify the epidemic curve data for the chosen forecasting model, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

- Web Services APIs: The My4Sight system architecture is shown in figure 4.2. The front end and back end are completely decoupled with each other, and data flows between them through carefully designed RESTful APIs. This design provides a lot of flexibility and allows the system to scale horizontally. We have developed several classes of APIs which provide access to various system resources and hide the complexity of the back end infrastructure. These APIs are meant for both internal and external client consumption. The My4Sight client in Figure 4.2 is one of the sample applications that is built on the top of these web services. These web services can be leveraged in the future to create more advanced or native applications for desktop and mobile platforms.

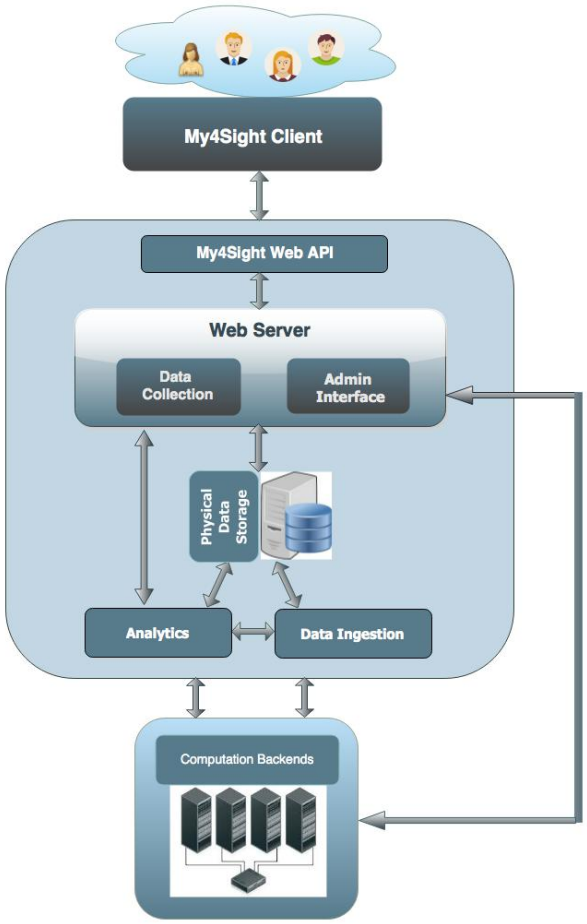


Figure 4.2: My4Sight System Architecture

- Deployment Time: The system reduces the amount of time and effort involved in launching new diseases, geographical regions, and forecasting models. Also, the framework provided by the system makes it easy and fast to deploy new web APIs.
- Analytical Engine: My4Sight is designed as an analytical platform, providing access to flexible queries, data collection features, and machine-learning algorithms. The machine-learning and algorithmic aspects are not the primary goal of this thesis, although they are very important. This component provides a generic framework where different algorithms could be evaluated in the future.
- User-Related Data: The system requires users to login in order to access the resources and complete human computation tasks mentioned in chapter 3. This helps the system to keep track of user participation and compute user related analytics. The system tracks the following information about each user:
  - Login IP address
  - GeoLocation data (latitude, longitude, city, state, country, etc.)
  - Login and logout time
  - Responses to human computation queries

  This information is stored in a relational database to manage and retrieve the data efficiently.

4.2 Implementation

The system has a modular architecture consisting of several loosely connected components built using open source tools. We chose to represent the system data in a relational format, as this provides us the ability to execute flexible queries based on a combination of geographical region, surveillance data source, forecasting method, etc.

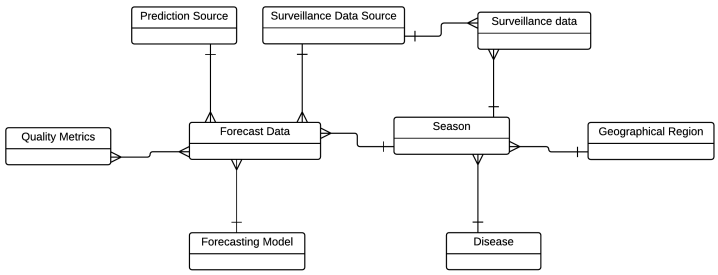


Figure 4.3: My4Sight Conceptual model. In this figure, only important entities and relationships are included.

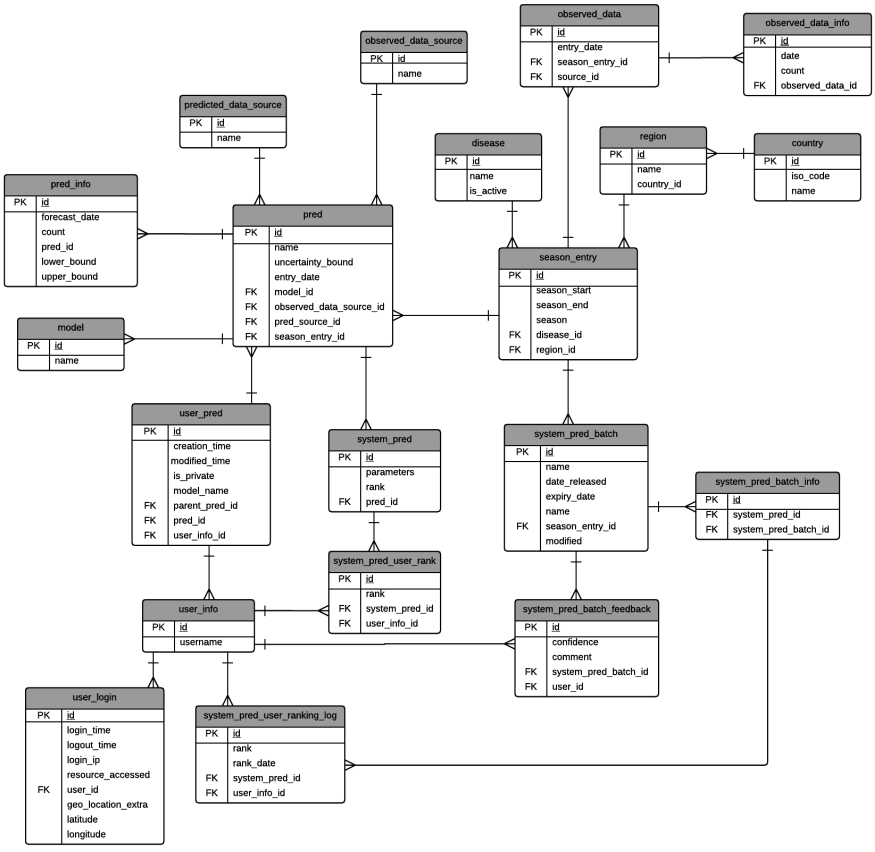


Figure 4.4: My4Sight database schema. The figure includes only relevant tables and relationships among them.

The conceptual model plays an important role in modeling the requirements of the system and represents various entities and the relationships between them. The model for My4Sight is shown in Figure 4.3. This model is kept simple by including only the relevant entities and relationships. In order to store and provide analytics for surveillance and forecasting data from past seasons, a "season" entity is used in the system. The idea of what constitutes a season is loosely defined to generalize the definition for multiple diseases. A season is identified by a combination of geographical region, disease, and season name. Generally, it is observed that data reported through traditional methods for illnesses such as flu lag by 2 to 3 weeks, and, further, that these reports are revised retrospectively after the first report. To account for this uncertainty associated with surveillance data, a single season entity can have multiple surveillance data sources. The database schema is shown in Figure 4.4.

4.2.1 Back End

The back end is written completely in Python and uses Django [9], a high-level Python web framework, to provide the server component. The reasons for choosing the Python ecosystem (Django can be viewed as Python extension) are that it encourages rapid application development and has excellent libraries for scientific programming, data statistics, and data manipulation [29]. Thus, the Python ecosystem provides the necessary "lego blocks" so that we can reuse existing components and focus our time on developing the desired features for the My4Sight back end. Currently, My4Sight uses a MySQL database for its storage needs, although this could easily be configured to use other back end databases like PostgreSQL or Oracle. See Figure 4.5 for the technology stack used in the development of the My4Sight back end.

Figure 4.5: Technology stack [15, 16, 9, 14, 13, 17] used in the My4Sight back end. Sources: http://www.mysql.com/, https://www.python.org/, https://www.djangoproject.com/, http://www.django-rest-framework.org/, http://www.celeryproject.org/, https://www.rabbitmq.com/. Used under fair use, 2015

Web Services API

The RESTful APIs are designed to be flexible, and provide access to system resources and data analysis methods. Figure 4.6 shows an example of the hierarchical API view as seen by a client application.

The APIs are designed using best RESTful patterns, and include the following features:

1. Multiple versions for backward compatibility

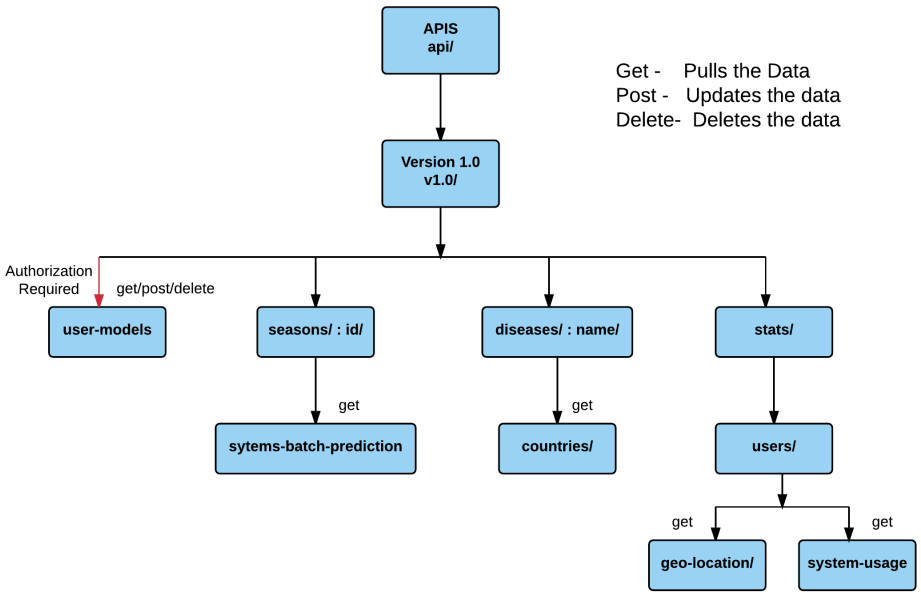


Figure 4.6: My4Sight API consumer view. The red line for the "user-models" node indicates that the API endpoint containing that node requires authentication.

2. Multiple representations/formats for system resources like prediction data, surveillance data sources, etc,
3. Each system resource is identified by a unique URL; see Figure 4.7 for an example.
4. Authentication for secure system resources
5. Queries, like filtering and selection, are built into the URL as query parameters. For an example, see Figure 4.7.

Since every resource that needs to be retrieved from the system is uniquely identified by a URL, the responses to APIs can be cached at multiple levels (server-side caching and client-side caching) to improve the system's scalability and efficiency. The My4Sight client (i.e., front end application) and back end employ caching to prevent the server and database from being hammered by repeated requests.

4.2.2 Front End

User experience plays an important role in human computation systems. In the case of My4Sight, this is no different as the system needs to provide visualizations of machine forecasts in order to collect human insights regarding epidemics. The My4Sight client, running

Service End Point	Method	Functionality
my4sight/api/v1.0/diseases/flu/countries	GET	Gets all the countries (along with their regions) where flu is currently prevalent
my4sight/api/v1.0/user-models/	POST	Creates a new user generated model
my4sight/api/v1.0/user-models/?func=list&limit=2&offset=0&ordering=model_name	GET	Get first two user models after sorting by “model name” in ASC order

Figure 4.7: Examples of My4Sight API endpoints and their descriptions

on the browser, needs to provide rich interactions to the user to support ranking and individualized forecasting tasks as described in chapter 3. For this purpose, the My4Sight client uses Single Page Application (SPA) [40] architecture in creating the front end website.

Single Page Application (SPA)

SPAs are a modern way of creating rich and responsive web applications, and aim to bring desktop-like user experiences to the browsers. In traditional web applications, whenever a user clicks on a link or button, the browser sends a request to the server. The server responds to the request by constructing a completely new page, and this is displayed to the user. However, in SPA, a website is loaded for the first time by fetching the necessary resources, like CSS, javascript, and HTML. After the first page load, only the requested information is fetched from the server on demand, and partial pages are redrawn on the client side. This mechanism reduces server round trips to construct a full HTML page as seen in traditional web applications, thereby providing an enhanced user experience.

Figure 4.8 shows the architecture of the My4Sight client.The architecture includes the following components:

- Services: Responsible for communicating with the back end using web services APIs.
- Models: Maintains user- and system-related data at the client side
- View: Responsible for displaying the data to the user
- Controller: Handles user interactions and coordinates the communication between models and services.

In order to handle the challenges involved in implementing a SPA application, we chose AngularJS [33], a client-side javascript framework. Figure 4.9 shows the complete software stack used in the My4Sight client development.

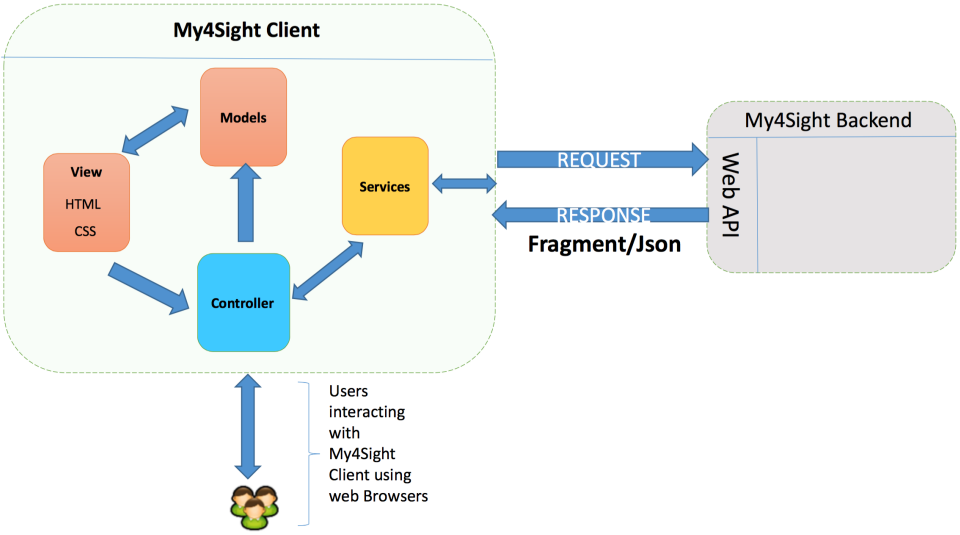


Figure 4.8: My4Sight Front End Architecture

Figure 4.9: Technology stack [19, 24, 18, 23, 25, 27, 22, 20, 26, 21] used in My4Sight Client. Sources: https://angularjs.org/, http://earthintegrate.com/10-reasons-why-javascript-is-the-best-language-for-the-web/, https://angular-ui.github.io/, https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5, https://jquery.com/, https://www.drupal.org/project/lodash, http://www.yourbusybee.com/using-fontawesome-icons-form-fields/, http://heyba.by/css5-logo/, http://leafletjs.com/, http://getbootstrap.com/, Used under fair use, 2015

Chapter 5

My4Sight Pipeline and Features

In previous chapters [3 and 4], we described the functionalities of the My4Sight front end and back end. In the current chapter, we will describe how the My4Sight pipeline is composed using the front and back ends to provide a complete, end-to-end, web-accessible system, and the role played by each component in the pipeline. Finally, at the end of this chapter, we will describe some of the features provided by My4Sight.

5.1 Pipeline

The My4Sight pipeline is graphically visualized in Figure 5.1. The various stages involved in the pipeline are numbered from 1 to 7. In the following sections, although we describe these stages in the context of forecasting models generated for a particular disease, geographical region, and season, the pipeline would be identical for any other combination of epidemic parameters since the system is built to support a variety of geographical regions, diseases, etc.

The tasks carried out in these stages include the following.

1. In the first stage, a set of forecasting model outputs are injected into the system using the data ingestion component. The process of generating different forecasting models is external to the system, and it is assumed that these models are likely generated using compartmental, agent-driven, statistical, and data-driven methods. The only constraint for these forecasting models is that they should predict the epicurve. The system is flexible enough to handle large numbers of forecasting models and each model, if applicable, can also have uncertainty bounds. The models could be injected by a requester/web administrator using scripts which have the necessary logic to do the translation and transformation into the My4Sight system data format. However, in

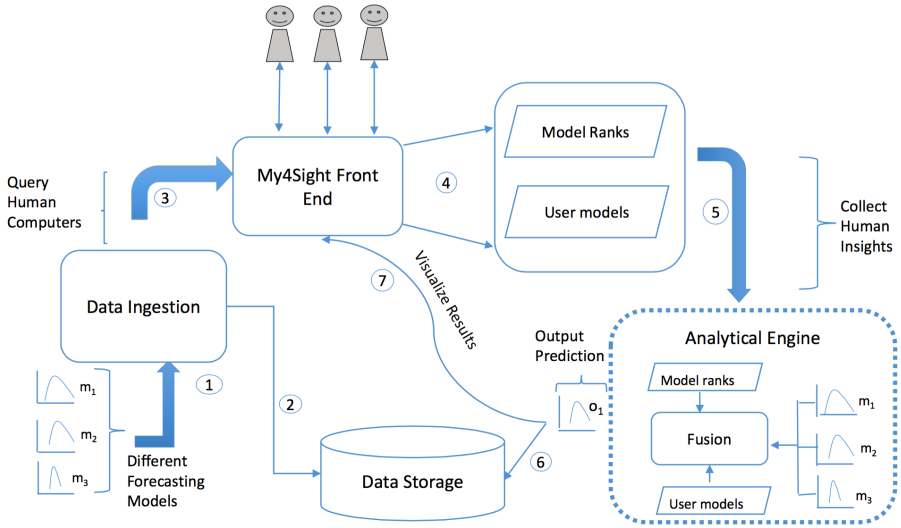


Figure 5.1: My4Sight System Pipeline

the absence of any such scripts for the new forecasting models, the user could use the web interface to manually add the data into the system.

2. In the second stage, the model outputs are stored in a relational format in a permanent database. This format allows us to perform flexible queries in an efficient manner. For example, sample queries may include, "Find the mean and standard deviation of all forecasting models for a particular geographical region and disease", "Find the distribution of the peak date predicted by different forecasting models", and "Find all the user forecasting models which are different from the system-generated predictions given a certain measure of dissimilarity".

3. In the third stage, the user selects a "disease" and "geographical region" after logging into the My4Sight client. This action from the user triggers the front end to communicate with the back end to retrieve the forecasting and surveillance data outputs, and previous responses corresponding to the user's selection, by using RESTful APIs provided by the My4Sight back end.

4. In the fourth stage, the forecasting data pulled from the back end is cast as two human computation tasks (ranking and individualized forecasting tasks) for users. If the user has already provided his response to the task, the front end visualization widgets are designed to reflect this state. This allows the user to evaluate his earlier response and provide a new response in the case of new evidence. The collected human insights are then stored in the back end database for later data transfer into the analytical engine.

5. In the fifth stage, collected human insights from all users, i.e. outputs of the ranking

and individualized forecasting tasks, are fused with system-generated predictions (the models provided as inputs to the system in stage 1) using machine learning and data analytical methods to produce a final forecast. For these fusion algorithms, the analytical engine could take the user's previous performance into consideration. Although these algorithms constitute an important part of the system, the focus of this thesis is on building a modular architecture and platform where different algorithms can be evaluated in the future. In the My4Sight pipeline depicted in Figure 5.1, all the components and stages are developed apart from the analytical component depicted by dotted lines. To illustrate how new algorithms could easily be added to the system, we have implemented a simple ensemble average for a set of predictions. These predictions could be system generated predictions or the ones selected by the user.

Ensemble Simple Average

Consider a set of predictions $y_1, y_2, \ldots, y_i, \ldots, y_K$ where K represents the total number of predictions, and

$$y_i = \{y_{t,i}\}_{t=1}^{n}$$

where n represents the epidemic outbreak period, which is unknown. Then, the ensemble simple average y_o is given by:

$$y_o = \sum_{i=1}^{K} w_i y_i = \{\frac{1}{K} \sum_{i=1}^{K} y_{t,i}\}_{t=1}^{n}$$

where $w_i = \frac{1}{K}$

6. In the sixth stage, the final forecast y_o generated by the analytical engine is stored in the back end database, along with the timestamp.

7. In the seventh stage, the resulting output is displayed to the user using appropriate UI widgets in the My4Sight client. For an example, see Figure 5.2.

5.2 Features

In addition to the pipeline described above, the system tracks user-related events, and GeoLocates users based on their IP address. The system is capable of running statistical analysis on the user-related data. The results are communicated to the user, wherever applicable, in the form of visualizations using the My4Sight client application.

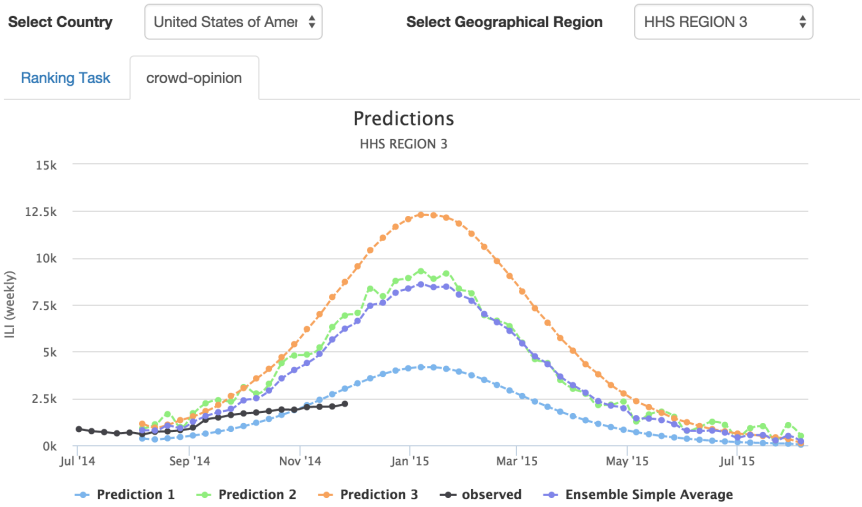


Figure 5.2: Simple Ensemble Average, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

GeoLocation

The GeoLocation information collected by the system is plotted on a high-resolution interactive map using markers. For an example, see Figure 5.3. This feature helps to analyze how the participants are dispersed geographically.

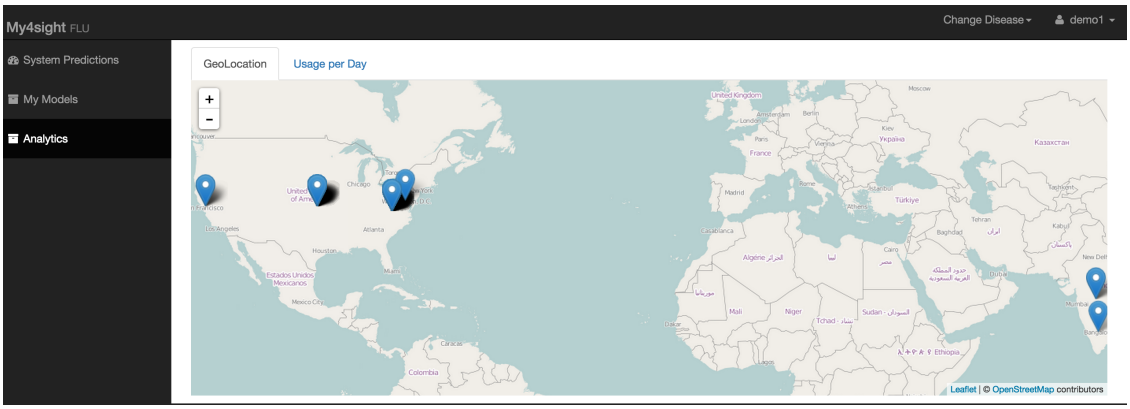


Figure 5.3: My4Sight GeoLocation feature, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

System Usage Statistics

User participation and contributions are a central component of My4Sight; therefore, the system provides tools to monitor their usage. These tools (see Figure 5.4) help in providing preliminary evidence for the following queries:

- How effective are the new incentive mechanisms and analytical features in motivating and retaining user participation over time?
- Does communicating with users via email and social media have any impact on system usage?
- What happens to system usage after the targeted campaign ends? It is very important that the system provides a value that is greater than the amount of time the user spends providing useful computation to the system.

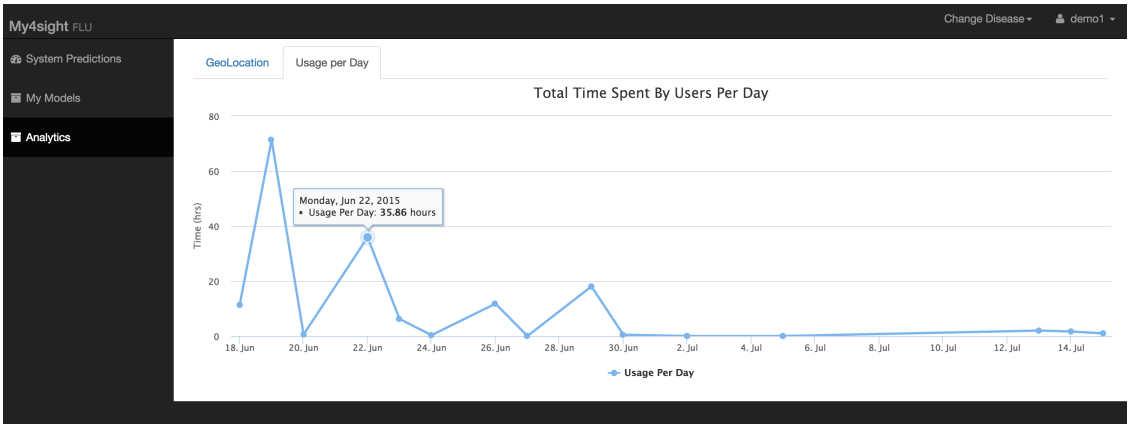


Figure 5.4: My4Sight system usage. The figure shows the total number of hours spent by users per day, http://epics.vbi.vt.edu/my4sight/, Used with permission of Dr. Madhav Marathe, 2015.

Chapter 6

Conclusion and Future Work

We presented My4Sight, a human computation system to integrate human insights with machine-generated forecasts to develop better predictive models. The system simplifies the addition of new diseases, forecasting models, geographical regions, and other dimensions of epidemics. The novelty of My4Sight is the development of a platform having the ability to quickly deploy, and seamlessly integrate, new algorithms for fusing human experiential knowledge with system-generated predictions. Our system spares users from worrying about the complexity involved in collecting human insights, understanding and developing the pipeline, and the framework to visualize the results. We discussed some of the user analytics features provided by the system. In our opinion, the data collected by the system would be invaluable for data-driven algorithms. With My4Sight, we hope to attract a lot of users who can provide useful insights about epidemics through ranking and individualized forecasting tasks. The system, with its infrastructure capabilities and features, provides a powerful toolbox to the community for combining human and machine intelligence into epidemiological forecasts.

The My4Sight system is still a work in progress. The future areas of work include:

1. Exploration of various methods and algorithms for combining human-generated outputs with machine-generated forecasts.
2. Modeling the performance and expertise of users over time and explore various incentive mechanisms for retaining the user base. One of the possible works in this direction could be assigning virtual points to the users based on their past performance, and a dashboard showing the users with the highest number of points in order to drive competition among users.
3. Development of additional complex queries to support the needs of analysts.
4. Identification of additional methods to aggregate human computation outputs generated via ranking and individualized forecasting tasks.

Bibliography

[1] Delphi — developing the theory and practise of epidemiological forecasting, . URL http://delphi.midas.cs.cmu.edu/.

[2] Epicast — epidemiological forecasting by delphi, . URL http://epicast.org/.

[3] Duolingo, . URL https://www.duolingo.com/.

[4] Duolingo - wikipedia, the free encyclopedia, . URL https://en.wikipedia.org/wiki/Duolingo.

[5] As the flu spreads across the country, Boston declares an emergency. URL http://www.consumerreports.org/cro/news/2013/01/as-the-flu-spreads-across-the-country-boston-declares-an-emergency/index.htm.

[6] WHO | Influenza. URL http://www.who.int/mediacentre/factsheets/2003/fs211/en/.

[7] Amazon external question type, . URL http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_ExternalQuestionArticle.html.

[8] Amazon Mechanical Turk, . URL https://www.mturk.com/mturk/welcome.

[9] The Web framework for perfectionists with deadlines | Django. URL https://www.djangoproject.com/.

[10] Flu Near You. URL https://www.flunearyou.org.

[11] Foldit, solve puzzles for science. URL https://fold.it/portal/.

[12] Crowdsourcing: Crowdsourcing: A Definition. URL http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html.

[13] Homepage | Celery: Distributed Task Queue, . URL http://www.celeryproject.org/.

[14] Django REST framework, . URL http://www.django-rest-framework.org/.

[15] MySQL :: The world's most popular open source database, . URL http://www.mysql.com/.

[16] Welcome to Python.org, . URL https://www.python.org/.

[17] Rabbitmq, . URL https://www.rabbitmq.com/.

[18] AngularUI for AngularJS, . URL https://angular-ui.github.io/.

[19] AngularJS — Superheroic JavaScript MVW Framework, . URL https://angularjs.org/.

[20] Bootstrap · The world's most popular mobile-first and responsive front-end framework., . URL http://getbootstrap.com/.

[21] Hello. It's Mai Ling... | A CSS3 badge logo, . URL http://heyba.by/css5-logo/.

[22] Using FontAwesome icons in Form Fields - YourBusyBeeYourBusyBee, . URL http://www.yourbusybee.com/using-fontawesome-icons-form-fields/.

[23] HTML5 - Web developer guide | MDN, . URL https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5.

[24] Why JavaScript is the Best Language for the Web, . URL http://earthintegrate.com/10-reasons-why-javascript-is-the-best-language-for-the-web/.

[25] jQuery, . URL https://jquery.com/.

[26] Leaflet - a JavaScript library for interactive maps, . URL http://leafletjs.com/.

[27] Lo-Dash | Drupal.org, . URL https://www.drupal.org/project/lodash.

[28] Pharmacovigilance App | Drug & Medication Side Effects Tracking. URL https://medwatcher.org/.

[29] NumericAndScientific - Python Wiki. URL https://wiki.python.org/moin/NumericAndScientific.

[30] Responsive Web Design. URL http://alistapart.com/article/responsive-web-design.

[31] Scicast. URL https://scicast.org/.

[32] Ushahidi. URL http://www.ushahidi.com/.

[33] AngularJS — Superheroic JavaScript MVW Framework. URL https://angularjs.org/.

[34] A model approach. *Nature*, 460(7256):667–667, August 2009. ISSN 0028-0836. doi: 10.1038/460667a. URL http://www.nature.com/nature/journal/v460/n7256/full/460667a.html.

[35] Google Flu Trends, November 2014. URL http://en.wikipedia.org/w/index.php?title=Google_Flu_Trends&oldid=631999740. Page Version ID: 631999740.

[36] CAPTCHA, June 2015. URL https://en.wikipedia.org/w/index.php?title=CAPTCHA&oldid=667304274. Page Version ID: 667304274.

[37] Flu season, May 2015. URL http://en.wikipedia.org/w/index.php?title=Flu_season&oldid=664936983. Page Version ID: 664936983.

[38] Influenza, June 2015. URL http://en.wikipedia.org/w/index.php?title=Influenza&oldid=665087181. Page Version ID: 665087181.

[39] Wikipedia:About, June 2015. URL https://en.wikipedia.org/w/index.php?title=Wikipedia:About&oldid=668633445. Page Version ID: 668633445.

[40] Single-page application, July 2015. URL https://en.wikipedia.org/w/index.php?title=Single-page_application&oldid=672381243. Page Version ID: 672381243.

[41] Epidemiology, June 2015. URL https://en.wikipedia.org/w/index.php?title=Epidemiology&oldid=668898583. Page Version ID: 668898583.

[42] Wikipedia:Size of Wikipedia, June 2015. URL https://en.wikipedia.org/w/index.php?title=Wikipedia:Size_of_Wikipedia&oldid=666942554. Page Version ID: 666942554.

[43] Armando Aguirre and Edilberto Gonzalez. The feasibility of forecasting influenza epidemics in cuba. *Memórias do Instituto Oswaldo Cruz*, 87(3):429–432, 1992.

[44] Eva Andersson, Sharon Kühlmann-Berenzon, Annika Linde, Linus Schiöler, Sandra Rubinova, and Marianne Frisén. Predictions by early indicators of the time and height of the peaks of yearly influenza outbreaks in sweden. *Scandinavian Journal of Public Health*, 2008.

[45] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. Soylent: A word processor with a crowd inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 313–322, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866078. URL http://doi.acm.org/10.1145/1866029.1866078.

[46] L.I. Besaleva and A.C. Weaver. Crowdhelp: A crowdsourcing application for improving disaster management. In *Global Humanitarian Technology Conference (GHTC), 2013 IEEE*, pages 185–190, Oct 2013. doi: 10.1109/GHTC.2013.6713678.

[47] Prithwish Chakraborty, Pejman Khadivi, Bryan Lewis, Aravindan Mahendiran, Jiangzhuo Chen, Patrick Butler, Elaine O. Nsoesie, Sumiko R. Mekaru, John S. Brownstein, Madhav V. Marathe, and Naren Ramakrishnan. *Forecasting a Moving Target: Ensemble Models for ILI Case Count Predictions*, chapter 30, pages 262–270. doi: 10.1137/1.9781611973440.30. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611973440.30.

[48] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski. Designing human friendly human interaction proofs (hips). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 711–720, New York, NY, USA, 2005. ACM. ISBN 1-58113-998-5. doi: 10.1145/1054972.1055070. URL http://doi.acm.org/10.1145/1054972.1055070.

[49] Kerstin Denecke, Peter Dolog, and Pavel Smrz. Making use of social media data in public health. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 243–246, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1230-1. doi: 10.1145/2187980.2188019. URL http://doi.acm.org/10.1145/2187980.2188019.

[50] Joshua M. Epstein. Modelling to contain pandemics. *Nature*, 460, 2009. doi: 10.1038/460687a. URL http://www.readcube.com/articles/10.1038%2F460687a.

[51] Stephen Eubank, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, May 2004. ISSN 0028-0836. doi: 10.1038/nature02541. URL http://dx.doi.org/10.1038/nature02541.

[52] J. Doyne Farmer and Duncan Foley. The economy needs agent-based modelling. *Nature*, 460, 2009. doi: 10.1038/460685a. URL http://www.readcube.com/articles/10.1038%2F460685a.

[53] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, February 2009. ISSN 0028-0836. doi: 10.1038/nature07634. URL http://dx.doi.org/10.1038/nature07634.

[54] David Alan Grier. *When Computers Were Human*. Princeton University Press, Princeton, NJ., September 2007. ISBN 9780691133829.

[55] Nattiya Kanhabua and Wolfgang Nejdl. Understanding the diversity of tweets in the time of outbreaks. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 1335–1342, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-2038-2. URL http://dl.acm.org/citation.cfm?id=2487788.2488172.

[56] Michael Kearns, Siddharth Suri, and Nick Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006. doi: 10.1126/science.1127207. URL http://www.sciencemag.org/content/313/5788/824.abstract.

[57] Patty Kostkova. A roadmap to integrated digital public health surveillance: The vision and the challenges. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 687–694, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-2038-2. URL http://dl.acm.org/citation.cfm?id=2487788.2488024.

[58] Anand Kulkarni, Prayag Narula, David Rolnitzky, and Nathan Kontny. Wish: Amplifying creative ability with expert crowds. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[59] Edith Law and Luis von Ahn. *Human Computation*. Morgan & Claypool Publishers, San Rafael, Calif., 1 edition edition, July 2011. ISBN 9781608455164.

[60] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data: Demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1474–1477, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487709. URL http://doi.acm.org/10.1145/2487575.2487709.

[61] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 68–76, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0222-7. doi: 10.1145/1837885.1837907. URL http://doi.acm.org/10.1145/1837885.1837907.

[62] Noelle-Angelique M. Molinari, Ismael R. Ortega-Sanchez, Mark L. Messonnier, William W. Thompson, Pascale M. Wortley, Eric Weintraub, and Carolyn B. Bridges. The annual impact of seasonal influenza in the US: Measuring disease burden and costs. *Vaccine*, 25(27):5086–5096, June 2007. ISSN 0264-410X. doi: 10.1016/j.vaccine.2007.03.046. URL http://www.sciencedirect.com/science/article/pii/S0264410X07003854.

[63] Elaine O. Nsoesie, John S. Brownstein, Naren Ramakrishnan, and Madhav V. Marathe. A systematic review of studies on forecasting the dynamics of influenza outbreaks. *Influenza and Other Respiratory Viruses*, 8(3):309–316, 2014. ISSN 1750-2659. doi: 10.1111/irv.12226. URL http://dx.doi.org/10.1111/irv.12226.

[64] Peter Organisciak, Jaime Teevan, Susan T. Dumais, Robert C. Miller, and Adam Tauman Kalai. A crowd of your own: Crowdsourcing for on-demand personalization. In *HCOMP'14*, pages –1–1, 2014.

[65] Philip M Polgreen, Forrest D Nelson, George R Neumann, and Robert A Weinstein. Use of prediction markets to forecast infectious disease activity. *Clinical Infectious Diseases*, 44(2):272–279, 2007.

[66] Miquel S Porta, Sander Greenland, Miguel Hernán, Isabel dos Santos Silva, and John M Last. *A dictionary of epidemiology*. Oxford University Press, 2014.

[67] Jeffrey Shaman, Virginia E. Pitzer, Cécile Viboud, Bryan T. Grenfell, and Marc Lipsitch. Absolute humidity and the seasonal onset of influenza in the continental united states. *PLoS Biol*, 8(2):e1000316, 02 2010. doi: 10.1371/journal.pbio.1000316. URL `http://dx.doi.org/10.1371%2Fjournal.pbio.1000316`.

[68] Jeffrey Shaman, Edward Goldstein, and Marc Lipsitch. Absolute humidity and pandemic versus epidemic influenza. *American Journal of Epidemiology*, 173(2):127–135, 2011. doi: 10.1093/aje/kwq347. URL `http://aje.oxfordjournals.org/content/173/2/127.abstract`.

[69] Patrick C. Shih, Kyungsik Han, and John M. Carroll. Community poll: Externalizing public sentiments in social media in a local community context. In *HCOMP'14*, pages –1–1, 2014.

[70] Kate Starbird. Digital volunteerism during disaster: Crowdsourcing information processing. In *Conference on Human Factors in Computing Systems*, 2011.

[71] James D. Tamerius, Jeffrey Shaman, Wladmir J. Alonso, Kimberly Bloom-Feshbach, Christopher K. Uejio, Andrew Comrie, and Cécile Viboud. Environmental predictors of seasonal influenza epidemics across temperate and tropical climates. *PLoS Pathog*, 9 (3):e1003194, 03 2013. doi: 10.1371/journal.ppat.1003194. URL `http://dx.doi.org/10.1371%2Fjournal.ppat.1003194`.

[72] Luis Von Ahn. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 418–419. IEEE, 2009.

[73] Jeff Yan and Ahmad Salah El Ahmad. Usability of captchas or usability issues in captcha design. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, SOUPS '08, pages 44–52, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-276-4. doi: 10.1145/1408664.1408671. URL `http://doi.acm.org/10.1145/1408664.1408671`.

Appendix A

List of Rest APIs

Service endpoint	my4sight/api/v1.0/diseases/flu/countries
method	GET
Description	Gets all the countries (along with their regions) where flu is currently prevalent

Table A.1: API to retrieve the list of countries where a disease is prevalent.

Service endpoint	my4sight/api/v1.0/user-models/
method	POST
Description	Creates a new user generated model

Table A.2: API to create a new user model

Service endpoint	my4sight/api/v1.0/user-models/?func=list
method	GET
Description	Retrieves user generated models
Limiting and Ordering Output	Query Parameters: limit, offset, ordering Ex: my4sight/api/v1.0/user-models/?func=list & limit=2 & offset=0 & ordering=mode _name Get first two user models after sorting by "model name" in ASC order

Table A.3: API to retrieve user generated models

Service endpoint	my4sight/api/v1.0/seasons/<id>/system-batch-predictions/latest/
method	GET
Description	Retrieves latest system generated predictions for a given season

Table A.4: API to retrieve latest system generated predictions for a given season

Service endpoint	my4sight/api/v1.0/stats/users/geo-location/
method	GET
Description	Retrieves geolocation data from all the logged in users

Table A.5: API to retrieve geolocation data from all logged in users.

Service endpoint	my4sight/api/v1.0/stats/users/system-usage/
method	GET
Description	Retrieves system usage statistics

Table A.6: API to retrieve system usage statistics.