An Intelligent Crowd-Worker Selection Approach for Reliable Content Labeling of Food Images

Mashfiqui Rabbi[1], Jean Costa[1], Fabian Okeke[1], Max Schachere[2], Mi Zhang[3], Tanzeem Choudhury[1]
[1] *Cornell University,* [2] *Hawken School,* [3] *Michigan State University*
ms2749@cornell.edu, jmd487@cornell.edu, fno2@cornell.edu, 15schmax@hawken.edu, mizhang@egr.msu.edu, tkc28@cornell.edu

ABSTRACT

Food journaling is an effective way to regulate excessive food intake. However manual food journaling is burdensome, and crowd-assisted food journaling has been explored to ease user burden. The crowd-assisted journaling uses a *label & verify* approach where an end-user uploads his/her food image and paid crowd-workers label content of the image. Then another set of crowd-workers verify the labels for correctness. In this paper, we propose an alternative approach where we label food images with only high performing labelers. Since high performing labelers generally provide good quality labels, our approach achieves high accuracy without verifying the food labels for correctness. We also propose a machine learning algorithm to automatically identify high performing crowd-labelers from a dataset of 3925 images collected over 5 months. Such automated identification of high performing workers and elimination of needless verification reduce cost of food labeling. Specially for large scale deployments where large number of images need to be labeled, our approach can reduce overall expenses by half.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

food journaling, crowd sourcing, health

1. INTRODUCTION

In 2007, the World Health Organization (WHO) declared obesity as a global epidemic [5]. A staple reason for obesity is excessive food intake. To prevent excessive food intake, past research explored food journaling as a way to self-regulate and change behavior [7]. Food journaling has attracted the Wireless Health community also and recent works explored the efficacy of food journaling with mobile phones [6]. However food journaling often requires high effort, since users need to manually search and log every food ingredient. To minimize the effort, researchers have explored the use of crowd-sourcing for food journaling where food pictures are labeled by workers from an online service (e.g., Amazon Mechanical Turk [1]).

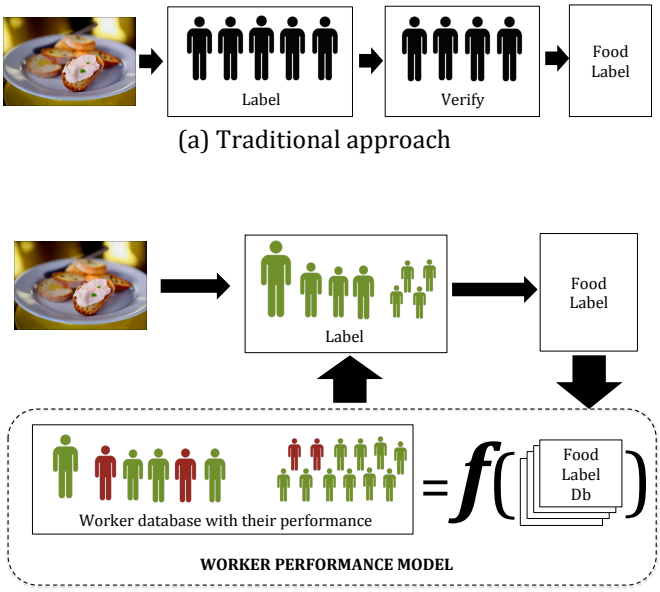


Figure 1: (a) Traditional methods doesn't distinguish between bad or good quality workers. These methods use paid workers to verify food labels to ensure good quality. (b) our approach keeps track of good quality workers and engages them only for good quality food labels.

In most traditional crowd-sourcing approaches to food label, each image is labeled by a set of paid workers. Then another set of paid workers review if the labels are correct. The most correct food labels are then defined as final food labels (Figure 1a). Such a method is oblivious to a worker's past history of how well s/he performed in labeling tasks. However, some workers are consistently good at labeling while some are bad. Therefore, it is not necessary to verify every food label by another set of paid workers since labeling quality is likely to be good if the image is labeled by high quality labelers. If costs for such verification workers are removed then the total end saving would be significant in large scale deployments where large number of images would be labeled.

Wireless Health '15 Oct 14-16, 2015, Bethesda, MD, USA

Given this insight, we propose a *worker performance aware* approach (Figure 1b), where we maintain a set of high performing labelers to gain high labeling accuracy. We also construct a machine learning model that can determine high performing labelers automatically. The model can look at the reply patterns of labelers and predict their food labeling performance with no cost. Specifically, the novel contributions of the paper are as follows:

- We construct a dataset of food images with crowd-worker provided labels. The dataset contains more than 3925 food images with 27784 food labels by crowd-workers and is collected by 16 users over 5 months long deployment (Section 2).
- From our dataset, we quantitatively demonstrate that it is possible to generate accurate labels for foods by only removing of low-performing workers (Section 3).
- We propose a machine learning model to distinguish between low and high performing labelers (i.e., how accurately the worker labels the food images). The model uses several characteristics from a worker's labeling patterns as features (Section 4). In addition, our model can detect low-performing workers at an early stage (Section 5).
- We implement an end user system that implements our model as a ready-to-use downloadable mobile application for Wireless Health researchers (Section 6).
- Finally, we demonstrate that our approach can yield accurate labels at a lower price than several baseline conditions (Section 7).

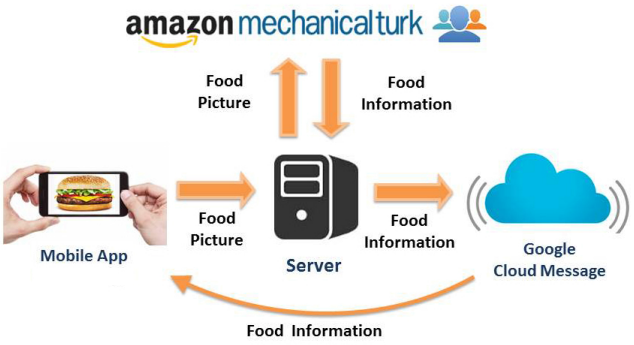


Figure 2: Food and calorie intake logging based on crowd sourcing using Amazon Mechanical Turk

2. DATASET

2.1 Mobile Application for Data Collection

We developed a mobile application for Android smartphones to log foods. Figure 2 shows the basic architecture of the application. A typical work flow of food logging using the smartphone application works as follows: first a user takes a picture of the food using the application (Figure 4a). The picture is then sent to a server and a Human Intelligence Task (HIT) is created at Amazon Mechanical Turk (AMT),

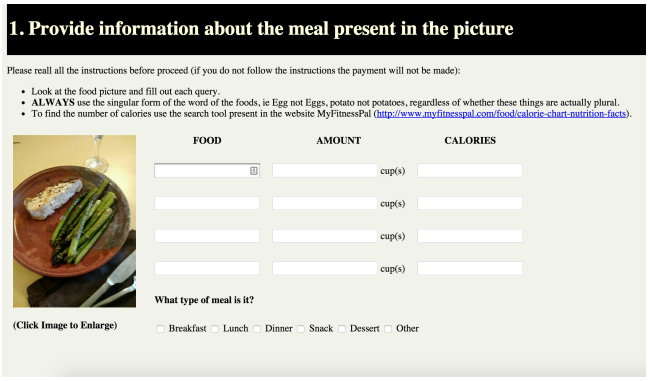
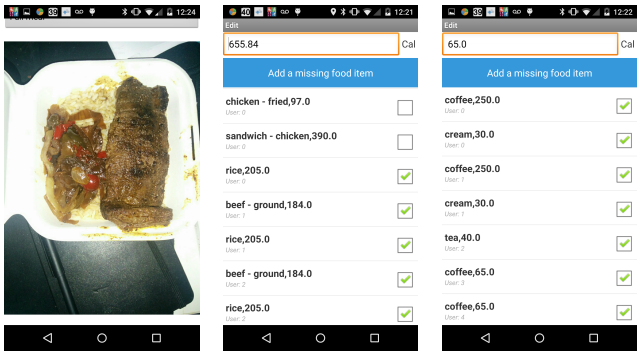


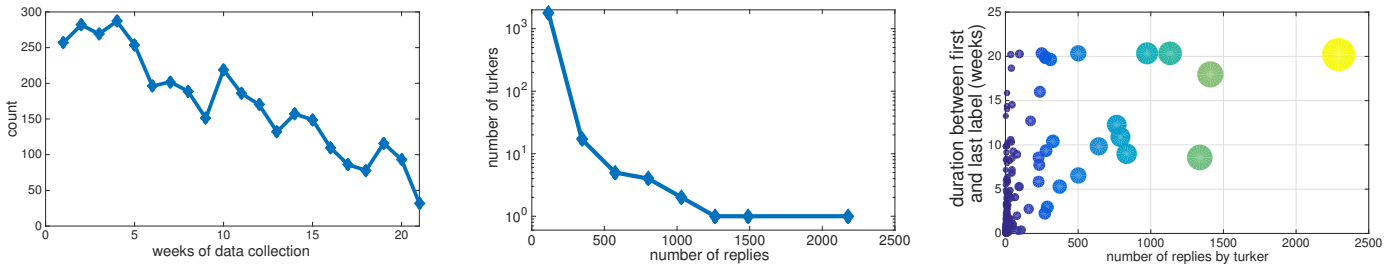
Figure 3: Food and calorie intake logging based on crowd sourcing using Amazon Mechanical Turk



(a) (b) (c)

Figure 4: (a)Taking food picture in the app (b) Replies from turker about the food image. Each replies consists of a food ingredient label and a corresponding calorie amount (c) Replies from turkers on a Starbucks coffee.

a popular crowd sourcing service. Then a set of crowd workers or ***turkers*** in AMT look at the picture, and provide the following information: food ingredient in the picture and an estimated calorie. A screenshot of the labeling interface that a turker uses to label a food is shown in Figure 3. In order to make the task easier, we implement an auto-complete feature - when the worker starts typing the food name, some suggestions of food appear in a window below the field. If the user selects a food, the corresponding calorie information per portion size is automatically added to the form. Then the turker only needs to update the food portion sizes and the number of calories is updated accordingly. The food suggestions in the auto-complete feature contain 50 popular foods from MyFitnessPal [8], a popular fitness application for phones. We decided to use this approach because larger databases contain many occurrences for the same food name which is confusing for the worker. Once the crowd-worker fills out all the required information, s/he earns 4 cents for completing the task, and the food information is sent and stored in the server. The server then sends the food labels and the calorie information to the Google Cloud Messaging (GCM). GCM then sends the food information to the Android device from which the food picture was taken. Figure 4 shows the application in action. As shown in the figure, the app lets users to remove wrong labels by unchecking them.



(a) (b) (c)

Figure 5: (a) Unique food images logged overtime during the data collection period (b) Distribution of turker response count. Large number of turkers replied to a few HITs. Small number of turkers replied to large number of hits (c) The distribution between turkers first and last response was often as high as 20 weeks. A more blue color and smaller size represents turkers replied less frequently. Frequently replying turkers often engaged with the food labeling processing for weeks.

Figure 6: A sample of images taken by users.

2.2 Dataset

Our dataset consists of 16 users who logged food using the smartphone application for 5 months. The data collection process resulted in 3925 food images captured with 27784 turker replies (each reply corresponds to one HIT). Figure 6 shows a collage of some images taken by users during the study. Figure 5a shows the counts of food images labeled across weeks during the study. 1801 unique turkers took part in labeling the food images. Figure 5b informs that the number of replies from turkers follow a heavy tail distribution, i.e. few turkers replied to a lot of requests while a large number of turkers replied to few requests. Finally, some turkers engaged with our system for extended period of time (Figure 5c).

Due to the significant amount of turker replies we received, it is hard to verify the quality of all 27784 turker replies. Therefore, we select a representative sample of turker replies and reason about different turkers' labeling performance. Specifically, we randomly selected 1200 replies out of 27784 replies and verified their quality as acceptable or not[1]. Two independent verifiers from our research team judged the labels provided by turkers for acceptability. If there is a disagreement with a label's quality then the third verifier was used to judge for acceptability. A turker label is considered acceptable if two or more verifiers considered the label to be acceptable. A similar approach to accept or reject turker replies was used by Thomaz et al. [11]. After the accept/reject process, 70% of turker labels are found acceptable. In the rest of this paper, the percentage of "acceptance" for a turker's replies is referred to as *ground truth accuracy*.

3. MOTIVATING WORKER-PERFORMANCE AWARE APPROACH

3.1 Accuracy of labeling

In our dataset, several frequently-replying low-performing turkers reduced the overall labeling accuracy significantly. Such turkers are shown in large blue dots in the lower part of Figure 7a where ground truth accuracy and frequency of replies for different turkers are plotted[2]. Overall 21% of total 27784 turker replies in our dataset are made by low performing turkers (with below <40% group truth accuracy). In the worst case, one of these low performing turkers replied nearly 5.7% of the total replies (Figure 7b). If such bad turkers are removed then cumulative labeling accuracy hits 85% with remaining turkers. i.e., less than 1 out of 5 labels can be wrong. Hence, it is unlikely a wrong label would be selected if 5 turkers are used to label a food image and majority voting [2] is used to decide final labels. Therefore, it is possible to get accurate labels by simply removing the low-performing turkers.

3.2 Cost Reduction

Given the importance of turker performance, how do we cheaply identify if a turker is doing well or not? The state-of-the-art relies on other paid turkers to evaluate the quality

[1]The labeling interface: http://goo.gl/uWabGy

[2]We use a similar color and sizing scheme for scatter plot figures in Figure 7 and 8a. The color of the dots represent ground truth accuracy of the turker. The color bar on the right represents this accuracy. e.g., yellow represents 100% ground truth accuracy while blue represents low accuracy. The size of the dot is proportional to how frequently a turker answers to the HITs.

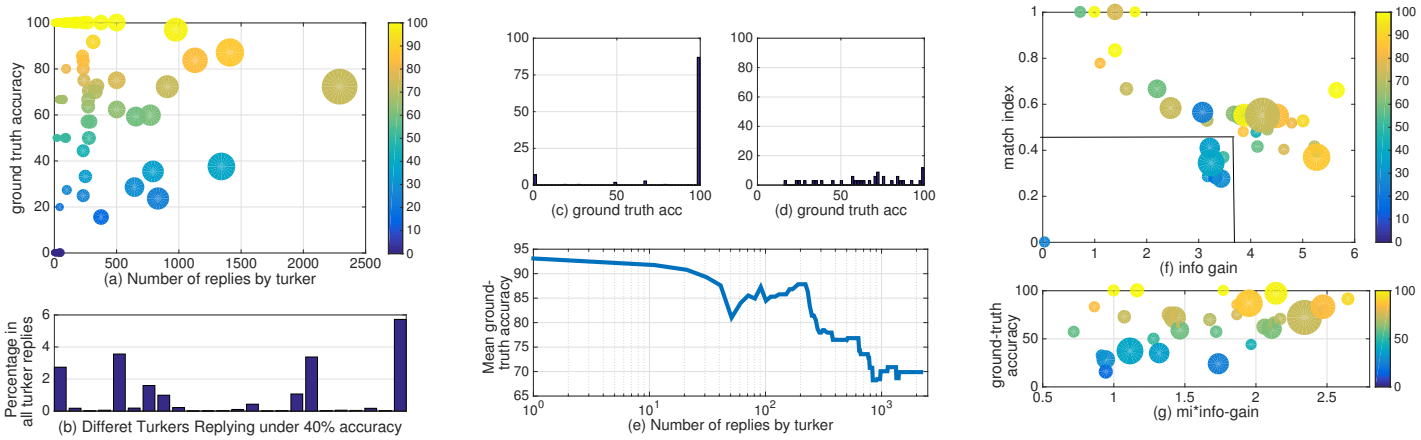


Figure 7: (a-b) Turkers replying to larger number of requests sometimes have low ground truth accuracy (c) Turkers who replied less than 100 HITs generally had high percentage correct labels (d) Ground truth accuracy varied widely for turkers who replied more than than 100 HITs (e) Cumulative accuracy as more higher replying turkers are added. Cumulative accuracy was nearly 85% for turkers replying less than 100 HITs (f) Distribution of match index and information gain (g) interaction of match index and information gain correlates with ground truth accuracy

of the labels [9]. If the quality is judged as bad, then the wrongly replying turkers do not receive payment. However from Figure 7a, we can see some turkers provide good labeling accuracy even when they replied to hundreds of HITs. On the other hand, some turkers provided bad quality labels consistently. Thus, good labeling performance can be achieved by engaging only high performing labelers during labeling and we can eliminate the verification of every label for correctness. Elimination of the verification would also reduce cost. In addition, we develop a machine learning algorithm to automatically detect turker performance/quality which can further reduce the cost.

4. A MODEL TO IDENTIFY WORKER PERFORMANCE

Reply patterns of high and low performing turkers are different. In this section, we propose a machine learning model that exploits the differences in reply patterns to predict turker performance. We first introduce some notations. Then we describe the features and machine learning model.

4.1 Notations

We denote the set of turkers as $T = \{t_1, t_2, t_3, ..., t_{|T|}\}$. The set of images labeled by the turker t_i is denoted as $P_i = \{p_i^1, p_i^2, p_i^3, ..., p_i^{|P_i|}\}$. Note here the same image can be labeled by multiple turkers. The replies of image p_i^j by turker t_i is denoted by set $r_i^j = \{r_i^j(1), r_i^j(2), r_i^j(3), ...\}$. For example, if a turker t_i replies to image p_i^j with steak, salad, and fries then $r_i^j(1), r_i^j(2), r_i^j(3)$ will be respectively "steak", "salad", and "fries". Finally, ground truth accuracy for turker t_i is denoted by y_i. Since replies from all turkers are not included in the subsample of 1200 replies in our dataset, y_i isn't defined for every t_i.

4.2 Features

We introduce several metrics that quantify the difference between high and low performing workers' reply patterns. We also illustrate the efficacy of the metrics with illustrative visualization and verbal explanation. We then use these metrics as features to build a machine learning model.

4.2.1 *Turker's Reply Frequency*

Infrequently replying workers often provide highly accurate labels whereas the frequently replying workers can be high or low performing. Figure 7c shows the distribution of ground truth accuracy of infrequently replying turkers that replied less than 100 times. Large percentage of these turkers replied with high accuracy. Such turkers constitute nearly 83% of our total 1801 turkers in the dataset with 13% of 27784 total replies. For the turkers with more than 100 replies that generated the remaining 87% replies, the ground truth accuracy is mixed with both high and low performances as shown in Figure 7d. Another way to look the same phenomenon is the changes in cumulative ground truth accuracy as more frequently replying turkers are considered (Figure 7e). Mathematically, the Y-axis corresponding to the X-axis or number of replies by a turker is $\frac{\sum_{i=1}^{|T|} y_i * |P_i| * \mathbb{1}_{[|P_i| \leq x]}}{\sum_{i=1}^{|T|} |P_i| * \mathbb{1}_{[|P_i| \leq x]}}$. i.e., we include turkers that replied less than x times and weight the ground truth accuracy with frequency of replies to give more importance to highly replying turkers. From Figure 7e, we see turkers with less than 10 replies and 100 replies, the cumulative ground truth accuracies are around 93% and 85%. However, as more high replying turkers are considered the accuracy drops to 70%.

4.2.2 *Match Index*

Replies from under performing turkers consistently do not match with replies from other turkers for same food images. However, the opposite is true for high performing labelers. Such differences happen because some labels are accurate for each image since 70% of the replies in the labeled dataset are acceptable. Thus on the same image, accurate labels from a high performing turker would consistently match with other accurate labels from another high performing turker. How-

ever for low-performing turkers, their wrong labels would not match with accurate labels from high performing turkers. In fact, wrong labels of low-performing turkers would not even match labels of other turkers since turkers can not coordinate. With this intuition, we calculate a *match index* MI_{ij} for an image p_i^j that a turker t_i replied to as follows: fraction of words in the t_i's reply to p_i^j that matches replies of other turkers to p_i^j. Before the matching, we preprocess the data with simple stemming from Natural Language Processing literature [3]. After stemming, 'Apple' and 'Apples' would be identical. We then compute a match index MI_i for turker t_i by computing the average of all match index measures MI_{ij}s. The exact equation of match index MI_i is as follows:

$$MI_i = \frac{1}{|Pi|}\sum_{j=1}^{|Pi|} MI_{ij}, \textbf{ and } MI_{ij} = \frac{\sum_{k=1}^{|r_i^j|} \mathbb{1}_{[r_i^j(k)\in r_i^{j'}]}}{|r_i^j|}$$

where $r_i^{j'}$ is defined as the set words replied by turkers other than t_i for image p_i^j. Intuitively, the above equation means if the words of a turker t_i's reply is contained in replies from other turkers to the same image then the MI_{ij} will be high. Now accurate replies from a turker has a high chance to match replies other high performing turkers. Thus a high performing turker will have high MI_{ij}s which means the average of MI_{ij}s denoted as MI_i will also be high.

4.2.3 Information Gain

Often under performing turkers use a small set food names for all replies. i.e., replies from under performing turkers lack diversity. On the other hand, replies from high performing turkers exhibit sufficient diversity since food labels would be different if different users eat different foods. We compute such diversity with an entropy or information gain measure in the following two steps (1) we first construct a dictionary D with unique words $\{d_1, d_2, d_3, ..., d_{|D|}\}$ contained in replies from all turkers. We pre-process the words using stemming techniques from natural language processing [3] (2) then we compute entropy with the following equation:

$$entropy_i = -\sum_{j=1}^{|D|} p_{ij} \textbf{ log } p_{ij}$$

$$\textbf{where } p_{ij} = \frac{\sum_{l=1}^{|P_i|}\sum_{k=1}^{|r_i^l|} \mathbb{1}_{[r_i^l(k)=d_j]}}{\sum_{l=1}^{|P_i|}\sum_{k=1}^{|r_i^l|} 1}$$

According to the above equation, if the food labels are predictable then entropy is low whereas if food labels are less predictable then entropy is high. Therefore, if turkers reply accurately then their replies would exhibit high diversity and their entropies would be high compared to low-performing turkers.

Figure 7f shows the distribution of match index and information gain. In the figure, the large blue dots representing highly replying bad quality turkers are largely located where both information gain and match indices are low. Figure 7g shows the relationship more prominently where ground truth accuracy positively correlates with the interaction [10] between match index and information gain ($r = 0.48, p = 0.001$).

4.2.4 End-User Acceptance Rate

Throughout the user study, end-users had the option to correct the labels provided by the turkers (see Figure 4(b-c)). We compute user acceptance rate of a turker as percentage of the turker's labels accepted by end-users. The user acceptance rate strongly correlates with ground truth accuracy ($r = 0.77, p < 0.0001$). Figure 8a shows the correlation of user acceptance rate with ground truth accuracy. However, user-corrections are relatively rare and only 1% (168/27784) turker replies were corrected by end users. Importantly, end-user corrections is more available for frequently replying turkers. 93% turkers replying more than 700 replies received some end-user corrections. In contrary, only 13% turkers under 700 replies received end-user correction.

4.3 Turker Performance Model

In this section, we formulate two machine learning regression models to identify turker performance or ground truth accuracy. One model uses the end-user acceptance rate feature and the other does not. The model without the user acceptance rate feature is as follows.

$$y_i' = \beta_0 + \beta_1 * MI_i + \beta_2 * entropy_i + \beta_3 * entropy_i * MI_i$$

where y_i' is the predicted ground truth accuracy. We add an interaction term between match index and information gain since such interaction correlates significantly with ground truth accuracy (Section 4.1.3 and Figure 7g). Our second model uses end-user acceptance feature. However, we can use end-user acceptance features for only high replying turkers where the feature is more available. The second model is as follows:

$$y_i' = \begin{cases} \beta_0 + \beta_1 * MI_i + \beta_2 * entropy_i \\ \quad +\beta_3 * entropy_i * MI_i \textbf{, where } |P_i| \le C \\ \\ \gamma_0 + \gamma_1 * MI_i + \gamma_2 * entropy_i \\ +\gamma_3 * entropy_i * MI_i + \gamma_4 * \text{user\_acc}_i \textbf{, otherwise} \end{cases}$$

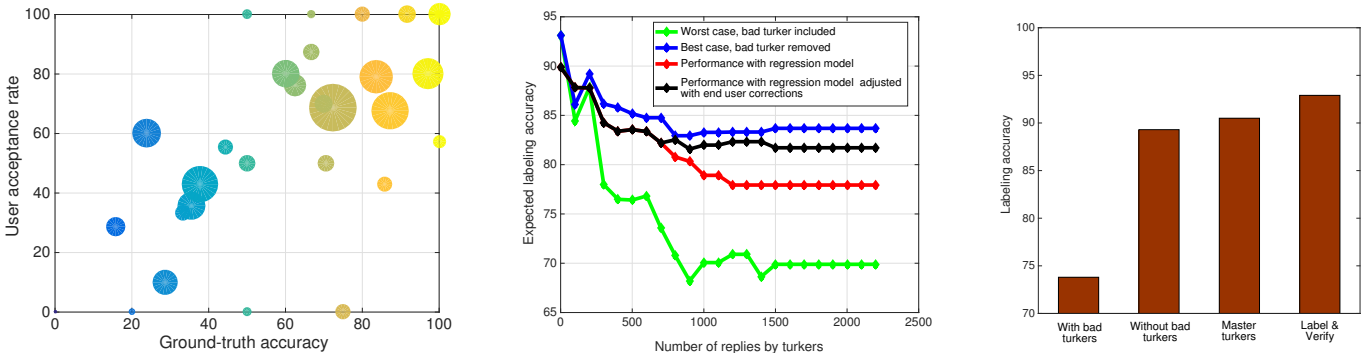
where a separate linear regression model is used with end-user acceptance feature only when the turker t_i replied more than some constant C number of times. Finally, we fit the above regression model in piecewise manner [4] for different turker reply frequency interval. We make the regression dependent of frequency because turker performance is found to depend on frequency in section 4.1.1. For a window size of 100 in frequency or $|P_i|$ (i.e., 0-100, 100-200, 200-300 and so on), we formulate a different linear regression.

4.4 Accuracy of Food Content Labeling

We evaluate the labeling performance of our models in two ways: (1) an offline evaluation where we count how many wrong labels are removed from our dataset if we discard low-performing turkers as predicted by our model; and (2) an online evaluation where we label new images by turkers that exclude low-performing turkers as predicted by our model. The accuracy of the labeling is then compared with several control conditions.

4.4.1 Offline Evaluation

We investigate the fraction of remaining accurate labels if labels of low-performing turkers from our model are excluded from our dataset. Such a measure would indicate labeling accuracy without bad-labelers in an offline setting. We do



(a) (b) (c)

Figure 8: (a) Labeling accuracy for turkers replying to larger number of requests have high correlation with end user acceptance. (b) Predicted accuracy model. If turkers are chosen based on predicted accuracy. Turkers with less than 60% accuracy are discarded. (c) Accuracy of an online deployment of our system.

the evaluation in two steps (i) we predict labeling performance of different turkers based on our model (ii) we estimate the percentage of remaining accurate labels after removing low-performing turkers from our model. To predict labeling performance of a turker t_i, we train a model with features from $T - t_i$ and predict the accuracy y'_i from the model for t_i. Such *leave-one-out* evaluation is commonly referred as cross-validation to evaluate machine learning models. Furthermore, the model trained over $T - t_i$ has zero knowledge of the replies and features of t_i. Thus the prediction for t_i indicates the way our model would perform for an unknown turker. Given y'_i, the quantity $y'_i * |P_i|$ would indicate the expected amount of accurate labels from t_i with our model. Specifically we examine a metric called *expected labeling accuracy*, which is defined as $\frac{\sum_{i=1}^{|T|} y'_i * |P_i| * \mathbb{1}_{[|P_i| \leq x]}}{\sum_{i=1}^{|T|} |P_i| * \mathbb{1}_{[|P_i| \leq x]}}$, to demonstrate efficacy of our model. Here $\mathbb{1}$ is a indicator function and the denominator is a normalization term. Intuitively, the expected labeling accuracy means the cumulative labeling accuracy we can expect if only turkers replying to x number of queries are considered.

Figure 8b shows the results. We first define a worst and best case of expected labeling accuracy for different values of x (i.e., number of replies by turkers). The green curve shows expected labeling accuracy with no bad turkers removed which is the worst case scenario for our model. The blue curve on the other hand shows a upper limit or best case, where we discard all turkers with 60%[3] or less accuracy (i.e., ground truth accuracy) known in the labeled training set. The red curve shows the expected labeling accuracy if we exclude turkers with a predicted accuracy of less than 60% using the model without user acceptance rate. As can be seen in Figure 8b, the model always performs better than the worst case. The expected labeling accuracy also stays around 83% if the number of replies from a turker is less than 700. However, after including turkers with 700 or more replies, the model performance deteriorates. A possible reason for deterioratation is the heavy tail nature of the turker reply distribution (Figure 5b) and there are not enough points to learn a reliable model with low number of turkers that replied higher than 700. However, if we use the model with user acceptance rate for $C = 700$ then we get the black line in Figure 8b. This model maintains accuracy of near 83% throughout where the possible upper limit model accuracy is around 84%.

[3]60% is chosen heuristically. Other accuracy numbers also achieve similar results

4.4.2 *Online Evaluation*

Due to the offline nature of the evaluation in last section, it is essential to know the performance of our model in an actual real-life deployment. In our online evaluation, we do so by deploying our model in real-life. We upload a set of 30 images and get labels for the images. These images are randomly selected from our dataset. For each image, we acquire labels from 5 turkers that do not include low-performing turkers as predicted by our algorithm. Since a proper evaluation needs comparison with baseline conditions, we evaluate against three conditions when (a) if bad turkers are not removed (b) against "master turkers", a list of high performing turkers in Amazon Mechanical Turk (c) with *label and verify* approach used in earlier work [9]. For label & verify, we recruited 5 turkers for labeling and 5 turkers for verifying. For each turker, we paid 0.04. For "master turkers", we could not get master turkers at $0.04. We paid $0.15 for each label from master turkers and recruited 5 master turkers for each food image. For each experiment scenario, a total of 150 labels are collected.

Two independent evaluators in the research team judged how many of these labels were acceptable. If there is a disagreement then a third evaluator judged the labels. Labels are considered acceptable if they are accepted by two or more times by the evaluators. Thomaz et al. [11] used a similar approach to accept or reject turker replies. Figure 8c shows percentage of acceptance. Acceptance after removing low-performing turkers with our method (88.7%) outperforms the control condition with no bad turkers removed (72.9%). Master turkers approach performed similarly to our approach (90.5%). Label and verify approach, where we considered food content labels that were accepted by majority of verification turkers, is 92.3% accurate. Although both master or label and verify approach performed with similar accuracy our approach, these approaches cost

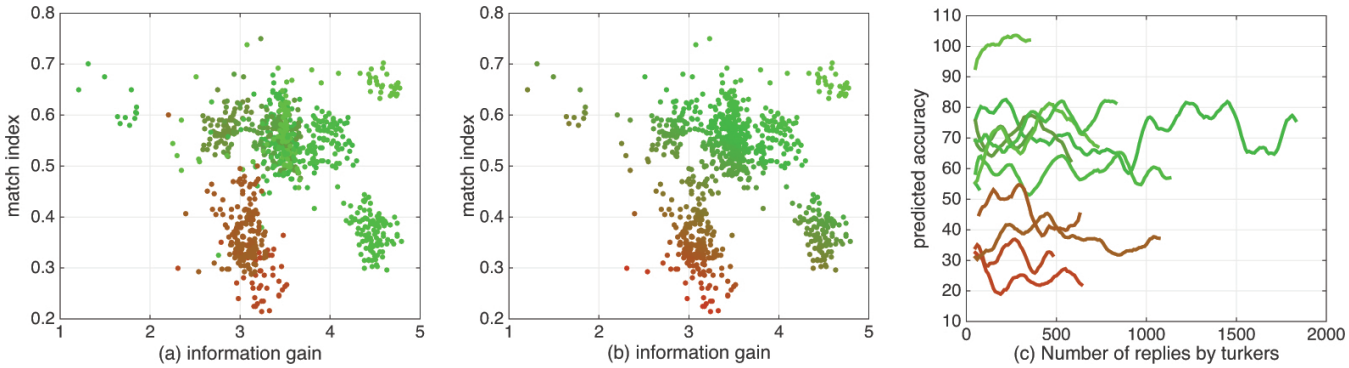


Figure 9: (a) distribution of match index and information gain on small windows of turker replies. Relatively red dots represent low quality turkers as defined by ground truth accuracy (b) same as 'a' but more red color of a dot is due to accuracy inferred by our model (c) evolution of inferred accuracy on small windows of turker replies.

significantly more than our approach. We will discuss the cost proposition in more detail in section 7.

5. EARLY DETECTION OF UNDER-PERFORMING WORKERS

So far, we built a model to identify a turker's performance from the turker's total history of replies. However an important problem is to detect low performing turkers early, so that they can be warned and eventually be removed before they adversely affect the labeling.

Previously discussed features also perform well for early detection of low-performing turkers. Figure 10a shows the match index and information gain for a window of 50 replies for turkers with more than 500 replies (i.e., both good and bad quality turkers)[4]. Dots from turkers with low quality replies are shown in more red colors (solid red is 0% ground truth accuracy) whereas a good quality turker is shown in more green color (solid green is 100% ground truth accuracy). The red dots, with low values of information gain and match index, are clearly separated from green dots. A linear regression model can learn these differences. Figure 10b shows the results of a 10 fold cross validation results (i.e., only predicted values are shown) of the linear regression model. Again we color code the dots similar to Figure 10a but with inferred accuracy rather than ground truth accuracy. We can see Figure 10b closely matches with Figure 10a.

In order to demonstrate early detection performance, Figure 10c shows the predicted accuracy on windows of 50 replies over time. Each curved line in the figure is a turker and more green lines represent high performing turkers (i.e., with high ground truth accuracy). A line representing low performing turker is shown in a more red line. Predicted accuracy in Y-axis at a point x on X-axis represent predicted accuracy on the window of $[x-24, x+25]$ replies. i.e., predicted accuracy at $x = 500$ is computed from 476th to 525th replies. In Figure 10c, more red color lines representing low-performing turkers consistently show low predicted accuracy (i.e., below 50% predicted accuracy) overtime. Thus within 50 replies, it is possible to understand which turkers might be low performing. Furthermore, the green lines representing high-performing turkers also show high predicted accuracy, thus a high-performing turker would not be flagged as low performing with a cut off of 50% predicted accuracy.

[4]We do not include turkers with less than 500 replies since the number of turkers are large and they are generally high performing. Thus including them in the visualization and discussion will crowd the figured unnecessarily. However, in our evaluation we get very similar results for turkers with less than 500 replies.

6. END-USER SYSTEM IMPLEMENTATION

Given the efficacy of our system, we implemented our algorithm as an end user system. The end user system consists of two components: (1) a smartphone application and (2) a web backend. The smartphone application and web backend are similar to what we already described in section 2.1.2. However, we added a module that computes the accuracy for each turker and uses the Amazon Mechanical Turk API to create a group of low performing turkers (who replied at least 100 times with predicted accuracy less than 60%). The low-perfomring turker group is avoided when new HIT requests are sent to mechanical turk. Also, the system can send warning to turkers if the reply quality drops in their last 50 replies.

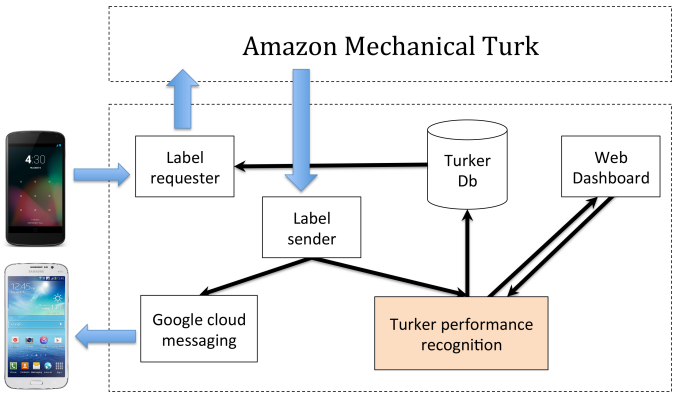


Figure 10: System architecture

Finally, the results from predictive model can be accessed via a dashboard. The dashboard also gives user control to ban or warn a under performing turker that may override the predictive model. Manual user control is specially useful if the predictive model makes a mistake. A running imple-

mentation of this dashboard along with the dataset and a functioning application can be accessed from here[5].

7. COST ANALYSIS

Our system can identify turker performance and early detect bad turkers. However, how much does our system cost to label foods? Furthermore, is the end accuracy acceptable? Our system requires $0.2 to label a image where we recruit 5 turkers, at $0.04 each. According to section 4.3.2, 88.7% of these turker labels are accurate. However if we deploy traditional *label and verify* [9] then we can get an accuracy of 92.3%, although it requires an extra $0.2 per image to filter wrong labels with additional turkers at $0.04 each. Finally, if we use master turkers - a set of high quality turkers maintained by Mechanical Turk - then we get 90.5% accuracy. However, assigning master turkers requires a steep price of $0.15 per label [11] with $0.75 per image.

The small accuracy loss would not affect end accuracy significantly, though our method costs significantly less. Our method incurs 50% and 73% less than all verify and master turker based approach respectively. We do so with an 1.8% less accuracy than master turkers and 3.6% less accuracy than earlier work [9]. The end effect 1.8% and 3.6% accuracy loss is minimal given we use majority voting [2] to decide final labels for an image: if 88.7% of turker labels are accurate then may be only 1 out of 5 labels received for an image can go wrong. Thus if we use a majority vote to select right labels then the wrong labels will not be likely included in the final set of labels. Therefore our approach can create accurate labels with 50% or less cost than competing techniques.

8. LIMITATIONS AND FUTURE WORK

Lack of Long Term Field Evaluation: Despite the efficacy of our system in online and offline evaluation, it is unknown on how our solution will perform in real life longitudinal trials. However, it is not be easy to work around our features. e.g., to get around information gain, a turker needs to consistently give garbage food labels that do not match with lavles to increase information gain. It is even harder to go around match index since wrong labels will not match with right labels by other high performing turkers.

Calorie Contents of Foods: In this work, we did not address the estimation of food calorie amounts. Measuring appropriate calorie amount from just food picture is unreliable, since it is hard to estimate size of foods from pictures. Furthermore, sometimes users do not consume the whole food in a picture. As a remedy, we give users control to adjust portion sizes. Then we use the portion size and calories per portion size to measure calories in the food.

Other Limitations: Although Amazon Mechanical Turk allows for incentivizing turkers, we do not incentivize high performing workers. It remains to be seen whether incentivizing the turkers increase turker performance even more. Furthermore, end user corrections are needed to reliably predict performance of frequently replying turkers. We can run similar correction step with other turkers and remove end-users dependence completely from the system. Finally, our approach advocates for selection and engineering of crowd-workers. Since crowd-workers are human labors, there are ethical concerns of our approach. However, we feel that our smart and cost-effective approach will create easier opportunities for larger scale data collection. Larger sized data then can enable completely automated food content labeling solution with machine learning that would not require crowd-worker manipulation.

[5]http://goo.gl/8B3E4e

9. CONCLUSION

In this paper, we introduced a crowd-source based food labeling system that produces high quality food content labels by exploiting crowd-workers' past labeling performance. Furthermore, we have constructed a machine learning model that can identify these high performing turkers and can early detect low performing turkers. Most importantly, our system costs significantly less than existing approaches. We open source our dataset and release a ready-to-use food journaling smartphone application. In our future work, we will work with experts and deploy the smartphone application in real world food journaling studies.

10. REFERENCES

[1] Amazon Mechanical Turk. http://www.mturk.com/, 2013. [Online; accessed 19 March 2015].

[2] J. P. Bigham, M. S. Bernstein, and E. Adar. Human-computer interaction and collective intelligence. 2014.

[3] S. Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[4] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[5] B. Caballero. The global epidemic of obesity: an overview. *Epidemiologic reviews*, 29(1):1–5, 2007.

[6] B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. M. Zhu, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey. Novel technologies for assessing dietary intake: evaluating the usability of a mobile telephone food record among adults and adolescents. *Journal of medical Internet research*, 14(2), 2012.

[7] E. Helander, K. Kaipainen, I. Korhonen, and B. Wansink. Factors related to sustained use of a free mobile app for dietary self-monitoring with photography and peer feedback: retrospective cohort study. *Journal of medical Internet research*, 16(4), 2014.

[8] MyFitnessPal, LLC. http://www.myfitnesspal.com/, 2013. [Online; accessed 19 March 2014].

[9] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 1–12. ACM, 2011.

[10] R. L. Ott and M. T. Longnecker. *An Introduction to Statistical Methods and Data analysis, 4th.* New York: Duxbury Press, 1993.

[11] E. Thomaz, A. Parnami, I. Essa, and G. D. Abowd. Feasibility of identifying eating moments from first-person images leveraging human computation. In *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference*, pages 26–33. ACM, 2013.