Compare Me Maybe: Crowd Entity Resolution Interfaces

Steven Euijong Whang, Julian McAuley, Hector Garcia-Molina
Computer Science Department, Stanford University
353 Serra Mall, Stanford, CA 94305, USA
{swhang, jmcauley, hector}@cs.stanford.edu

Abstract

We study the problem of enhancing entity resolution (ER) with the help of crowdsourcing. ER is the problem of identifying records that refer to the same real-world entity and can be an extremely difficult process for computer algorithms alone. For example, figuring out which images refer to the same person can be a hard task for computers, but an easy one for humans. An important component of crowdsourcing is the interface that is used for human and algorithm interaction. In this paper, we explore how the interface design along with other factors impact the human quality of comparing records. We also propose a model for separating good human workers from bad workers. Our analysis is based on extensive experiments on Amazon Mechanical Turk using real and synthetic image datasets.

I. Introduction

Entity Resolution (ER) is the process of identifying and merging records judged to represent the same real-world entity. Often, humans are better than a computer at determining if records represent the same entity. For instance, it may be hard for a computer to determine that the "Canon EOS Kiss X6i" camera is equivalent to the "Canon EOS Rebel T4i" camera (one is the Japanese-market name, the other the North American name). Similarly, to check if two items are the same, one may have to examine their photographs or user evaluations, something that is easier for humans to do. With the advent of platforms for human computation [1], [3], it is now much easier to use humans workers in the ER process.

There are many factors that can influence the quality of human comparisons. One of the most important factors is what we call the human *interface*, i.e., how the comparisons are presented to the worker and the choices she is given for replying. For instance, is a worker asked to compare several pairs of records or just one pair at a time? What data from each record is displayed, and how is it displayed? Is the worker asked for a simple Yes/No answer, or is she given an option to say she is not sure if the records match? Which of these strategies yields the best accuracy and lowest cost? Different types of interfaces have been used in various crowd ER studies [15], [17], [18], but we are not aware of any work that has thoroughly studied the choices, looking at factors such as the difficulty of the comparisons and the subject matter.

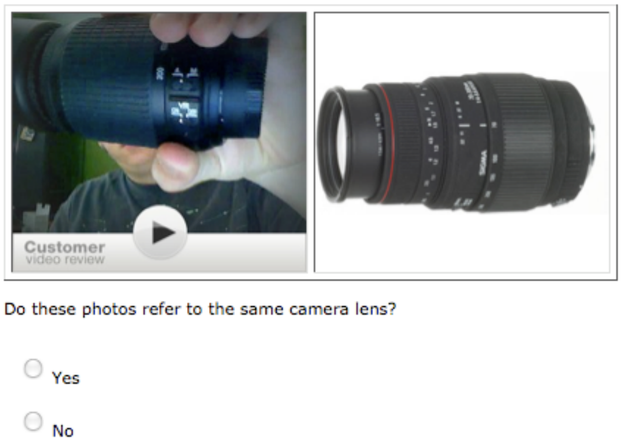
Our goal in this paper is to gain a better understanding of interface choices for crowd ER, including their benefits and drawbacks. However, as much as we would like to study all possible interfaces in all possible application domains, we clearly need to limit our scope. Hence, for our work we narrow the scope as follows:

- We focus on *pairwise interfaces* where each comparison involves two records only. To illustrate, Figure 1 shows a sample question that compares a pair of camera lenses. (A comparison where the worker is shown, say, 6 records, and asked to find the ones that match, would not be pairwise.) We believe that pairwise comparisons are simpler to deal with, both for the worker, as well as for the ER algorithm that generates the comparisons.
- We focus on comparisons of entity *images* (photos). That is, in our interfaces we will ask workers to compare images of two entities (e.g., people, products) and ask them if they represent the same real-world entity. Limiting ourselves to images significantly reduces the number of issues to consider. (For instance, if we were comparing products described by web pages, we would need to consider issues such as the page layout, ad removal, liveness of links on the page, and so on.)
- We focus on people comparisons, and include a couple of other domains (products, abstract images) just for comparison. We chose people comparisons because workers do not need special abilities or training, and because of the availability of data sets with gold standards.
- We focus on issues related to the interface, not on generic crowdsourcing issues. For example, worker training and testing is a very important factor for any crowdsourcing system, but it is mainly orthogonal to the interface, and hence not considered here. We do study detection of bad workers (sometimes called spammers), but only to see if different interfaces make detection easier or harder. (We do not try to determine the best overall technique for detecting bad workers.)

Even with our limited scope, there are many interesting questions to address. For instance, how does comparison difficulty impact worker accuracy, and what interfaces might be better suited for harder (or easier) tasks? And what makes people comparisons harder or easier? The age of subjects? Whether the subject is a famous movie star or a private citizen? Do the demographics of workers play a role in people comparisons? Does giving workers more options for responses (e.g., “the two people are somewhat similar”) yield more useful results for ER, or does it overwhelm the worker?

In summary, our contributions are as follows.

- We describe a crowd ER framework where the interface is an explicit component (Section II).
- We propose several interfaces for image comparisons, and study the impact of factors that may impact the interface



Do these photos refer to the same camera lens?

○ Yes

○ No

Fig. 1. Comparing two camera lens photos

effectiveness (Section III). For our evaluations we use actual results from Mechanical Turk workers, using real images (plus one experiment with synthetic images) (Section IV).

- Majority voting is often used to improve accuracy. (Several workers perform the same tasks, and the majority result is used.) We study the impact of the interface on majority voting (also in Section IV).
- We propose and evaluate models for identifying good and bad workers, again looking at the effect of the interface (Section V).

II. Crowd ER Framework

We first describe traditional ER and then explain how ER interacts with the crowd by asking questions. We then explain how questions may be replicated in order to improve accuracy.

A. ER Model

An ER algorithm E receives as input a set of records $R = \{r_1, \dots, r_n\}$ and returns a set of merged records R'. Each input record is assumed to refer to exactly one entity. Figure 2 shows how an ER algorithm can interact with the crowd. Based on the similarities among records, the *ER module* may decide which questions to ask to humans. The *Replicate module* optionally duplicates each question and can aggregate the different answers for that question (in order to improve accuracy). A task is a unit of work that is done by a worker. The *Interface module* creates tasks that contain questions and displays them through an interface. Finally, the worker answers for the questions are reflected into the ER process.

A recent work [18] focuses on the ER module where the problem is to choose the most "useful" questions to show to the human that can help improve the ER accuracy. Since human workers are relatively expensive (e.g., a few cents per several

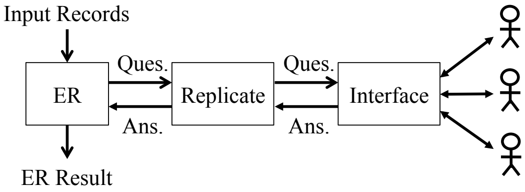


Fig. 2. ER using the crowd

record comparisons), only a limited number of questions can be asked. By carefully choosing questions and using blocking techniques [10], it is possible to significantly reduce the number of questions to ask for resolving large datasets.

The above work makes the assumption that the workers always answer questions correctly. In this paper, we do *not* assume that humans are perfect and focus on improving the Interface module. We define the *comparison accuracy* as the fraction of worker answers that are correct. For example, if 6 out of 10 responses are correct, the comparison accuracy is 0.6. Given a fixed budget for asking questions, we would like to find the best way to interact with humans and maximize the comparison accuracy.

B. Replicating Questions

Majority voting is a popular and effective way of filtering out bad responses and improving the comparison quality. The idea is to repeat the same question and take the majority of the answers. For example, suppose we asked a question 9 times. If 5 of the answers are Yes, we consider the final answer to be a Yes. Otherwise, we return a No. The Replication module in Figure 2 implements majority voting by repeating questions and aggregating the answers for each question.

We analyze how many times we need to repeat a question in order to arrive at a high accuracy. Suppose that all the questions asked have a probability p of being answered correctly and $1 - p$ incorrectly. In addition, we assume that all questions are independent of each other in terms of being answered correctly. Say that we ask each question $2 \times k + 1$ times. If we choose the majority response for each question (i.e., for each question we choose the answer that occurs at least $k + 1$ times), each question has a probability $\sum_{i=k+1 \ldots n} \binom{n}{i} p^i (1-p)^{n-i}$ of being answered correctly. For example, if $p = 0.7$ and $k = 4$, then the expected accuracy is 0.901, so there is a $0.901 - 0.7 = 0.201$ accuracy improvement.

III. Interfaces

We first study a basic pairwise interface (called Y/N) that requires a Yes or No answer. We then extend the Y/N interface by adding a new option called Maybe. In Section IV-E, we also consider variants of the two interfaces.

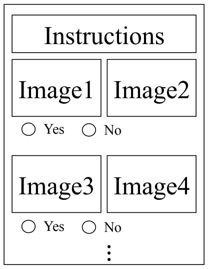


Fig. 3. Y/N interface format

A. Y/N Interface

The Y/N interface for a task consists of two parts (see Figure 3). On the top of the screen are instructions for comparing the entity images. For example, we can describe the purpose of the ER project and provide examples for comparing the records. The interface then lists one or more pairs of records along with Yes or No options. For example, Figure 1 shows a screenshot of the Y/N interface we used for our experiments when comparing two camera lens photos. The number of image pairs N per task can be adjusted based on how much work we want to assign per task. In our experiments (Section IV), we set N = 5 as a default.

While we study various aspects of the Y/N interface, we do not study every possible issue. For example, we do not vary the image size or the locations where the Yes and No radio buttons are displayed. Nor do we compare the vertical or horizontal placements for each image pair. Instead, we are mainly interested in how workers respond when they are required to only give a Yes or No answer.

B. Y/N/M Interface

In some cases, workers may not be sure whether two records are the same, no matter how skilled they are. By giving the option to express uncertainty, the Maybe option allows workers to avoid questions they are likely to answer incorrectly. Compared to the Y/N interface, we add Maybe as the third option in addition to Yes and No. We call this interface $Y/N/M$. Using the Maybe option has several potential effects on the comparison accuracy. The option can improve the comparison accuracy by filtering out potentially incorrect answers. For example, suppose that there are two questions q_1 and q_2 where the correct answers are both Yes (i.e., both pairs of records match). Say that one worker w_1 always answers correctly without using the Maybe option while another worker w_2 is not so confident and frequently chooses the Maybe option. If we provide the Y/N interface and w_2 answers Yes to q_1 and No to q_2, then the comparison accuracy is $\frac{2+1}{2+2} = 0.75$. On the other hand, if we provide the $Y/N/M$ interface and w_2 answers Maybe for both questions, then the accuracy increases to $\frac{2}{2} = 1$. The downside of the Maybe option is that there may be too few Yes or No answers. In this case, taking the majority of the few available answers may not be effective.

The Maybe option can be generalized to a scale of values with varying degrees of uncertainty. For instance, we can ask for an answer within a range of values [1, . . ., m]. The closer the answer is to m, the more the worker is leaning towards a match. An answer towards the middle of 1 and m indicates more uncertainty. While providing more options to workers encourages more informative answers, the workers could feel overwhelmed and provide meaningless information. In Section IV-E1, we compare the $Y/N/M$ interface with the generalized range interface and see which interface captures more information.

C. Possible Improvements

There are many possible improvements to the Y/N and $Y/N/M$ interfaces. For example, restricting the domain in the instructions can help the workers narrow down their decisions. Or adding training examples of matching and non-matching images can help workers perform better with more experience. Other improvements include deciding which difficult questions to ask more frequently based on previous answers and blacklisting bad workers. (See even more improvements in reference [13].) As we mentioned in Section I, it is impossible to evaluate every feature of an interface, so we do not cover these improvements in our experiments.

D. Other Interfaces

Until now, we have focused on two interfaces that compare records in a pairwise fashion. In this section, we explore possible extensions for the pairwise record format.

Pairwise Cluster Comparison: The worker determines if one record matches a set of records that are known to match. For example, we can now compare a cluster $\{r, s\}$ with a cluster $\{t\}$ where r and s are known to be about the same entity. Compared to the pairwise interface, we now show more information that can add more evidence towards a match or non-match. In our example, the worker has more information when comparing $\{r, s\}$ with $\{t\}$ than when she is comparing the individual records r with t. Notice that a pair of clusters can be constructed from a pair of records by augmenting the records with other matching records. Hence, the pairwise cluster interface is a straightforward extension of the Y/N and $Y/N/M$ interfaces. In Section IV-E2, we show that the additional information can indeed improve the comparison accuracy.

Mapping and Clustering: Going beyond pairwise record or cluster comparisons, we can display a set of records through the interface and ask workers to either map [15] or cluster [17] the matching records. Compared to a pairwise interface, the workers have more freedom in comparing records. In addition, there is more information because many records are compared at a time.

On the other hand, the flexibility raises multiple design questions. First, there is an issue of how many records to display for each question. Since the screen size is limited, we may not want to show too many records at a time. Second, selecting which records to compare is not obvious. One solution is to show records are likely to match with each other [17]. Third, there are many possible types of questions to ask to the workers. For example, instead of asking which records match, we could ask which records are definitely not the same as other records.

IV. Interface Evaluation

We perform extensive image comparison experiments on the pairwise interfaces that have been discussed. We compare the Y/N and $Y/N/M$ interfaces and identify the major factors that can influence the comparison accuracy. We then study other extended interfaces. Our crowdsourcing experiments were done on Amazon Mechanical Turk (AMT) [1].

Our goal is not to precisely predict error rates for photo comparisons, but rather to understand factors that influence worker accuracy. For example we would like to know if people comparisons yield higher quality results than product comparisons. As another example, does the introduction of the Maybe option lead to better comparison accuracy? Hence, we rely on shorter (and more numerous) exploratory experiments to identify trends. Also note that we cannot possibly be exhaustive in our exploration of image comparisons where there are thousands (if not millions) of types of photos and factors to consider. Hence, we chose a limited number of factors to study (e.g., the impact of the person's age, the type of sport, and so on). Even with our limited scope, we believe we study more image type variations than most other papers.

In the following sections, we evaluate the major factors on the comparison accuracy. In Section IV-A, we describe our experimental setup. In Section IV-B, we discuss how the interface design and majority voting impacts accuracy. In Section IV-C, we show how the difficulty of the task impacts the accuracy. In Section IV-D, we show how the worker ability impacts the accuracy. Finally in Section IV-E, we evaluate two other interfaces that extend Y/N and $Y/N/M$.

A. *Experimental Setting*

We explain the datasets used in our experiments and describe how we generate the tasks for AMT. Each task is called a Human Intelligence Task (HIT) where we can post questions on an interface. We also define two accuracy measures.

Datasets: Table I shows the image datasets used for evaluating our interfaces. The *Sports* dataset contains sports photos of Stanford athletes and consists of the following sports: gymnastics, baseball, softball, water polo, basketball, diving, field hockey, volleyball, and wrestling. All the athletes were labeled by name in each photo. The *Family* dataset contains family photos of an individual collected and labeled by his family members over many years. The *MovieStars* dataset contains photos

Dataset	Size	Description
People		
Sports	802	Sports photos of Stanford athletes
Family	9,678	Extended family photos of individual
MovieStars	18	Similar-looking movie star photos
Objects		
CameraLens	18	Camera lens photos
Abstract		
Dots	2M	Dot images

TABLE I

Datasets

of the following similar-looking movie stars: William Hurt, Jeff Daniels, and Ryan O'Neal [1]. For each movie star, we obtained three photos when the star was young and three recent photos when the star was older from an image search engine. The *CameraLens* dataset contains photos of three similar-looking camera lenses: the Nikon AF-S DX VR 55-200mm, the Nikon AF-S DX 18-300mm, and the Sigma 70-300mm. For each lens, we collected (again from an image search engine) three clear photos that show the specs of the lenses and three photos of poor quality where the lenses were harder to identify. Finally, the *Dots* dataset contains dot images. For each image, 1–1,000 dots are randomly spread in varying colors. We considered two dot images to be the same when they contained the same number of dots. For all the datasets above, we had gold standards that were used to evaluate worker performance.

Figure 4 shows four pairs of photos that we compared in our experiments. Photos 1 and 2 refer to the same gymnast. The different postures make it difficult to tell that the two athletes are the same person. Photos 3 and 4 refer to different baseball players. While the players are hard to distinguish with their similar uniforms and helmets, their jersey numbers (11 and 30) clearly show they are not the same person. Photos 5 and 6 show two different movie stars: Ryan O'Neal and Jeff Daniels. Even if they do not look very similar, the age difference makes it hard to conclude that they are different people either. Finally, photos 7 and 8 show a non-celebrity in different ages. The comparison is challenging because of the age difference and the fact that this person is not well known to the public.

HIT Generation: For the Y/N and $Y/N/M$ interfaces, our default HIT format was to display 5 pairs of images. At the end of the HIT, we also asked for any feedback on the HIT. In Section IV-E, we use other HIT formats where we compare clusters instead of records or ask for an answer within a range instead of a Yes or No. For each HIT, we payed 2 cents as a

[1] according to a site dedicated to finding similar-looking stars.

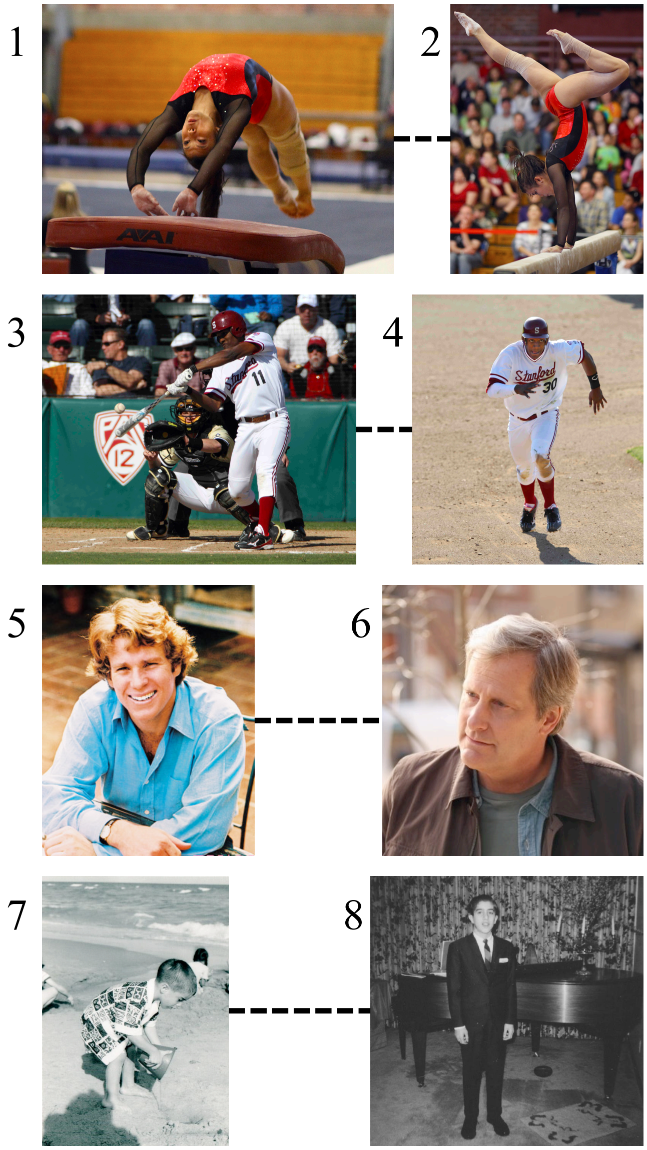


Fig. 4. Sample photo pairs

default.

We extracted pairs of photos from our datasets to create HITs. Since we could not compare all the images pairs exhaustively, we ran smaller experiments that compared a fixed number of image pairs. We did not select any photos at random since most comparisons would be non-matches. Instead, we wanted to have roughly the same number of matches and non-matches. From each dataset, we selected "challenging" comparisons where the images look similar. For example, we compared gymnasts in the *Sports* dataset who were wearing the same uniform, actors from the *MovieStars* dataset that looked alike, and identical people from the *Family* dataset at different ages. Hence, our accuracy numbers should be thought of as "worst case" for image comparisons.

Table II shows the various experiments we performed for comparing images. For the experiments $\mathcal{G}$, $\mathcal{B}$, and $\mathcal{S}$, we selected 8

Exp.	**Dataset**	**Pairs**
People		
$\mathcal{G}$	*Sports*	Exhaustive pairs of 3 gymnasts
$\mathcal{G}_{11}$	*Sports*	Random pairs of 11 gymnasts
$\mathcal{S}$	*Sports*	Exhaustive pairs of 3 softball athletes
$\mathcal{B}$	*Sports*	Exhaustive pairs of 3 baseball athletes
$\mathcal{F}$	*Family*	Random pairs of 3 family members
$\mathcal{M}$	*MovieStars*	Random pairs of 3 movie stars
Objects		
$\mathcal{C}$	*CameraLens*	Random pairs of 3 camera lenses
Abstract		
$\mathcal{D}$	*Dots*	Random pairs of 50 dot images

TABLE II

Experiments using data sets

athlete photos that referred to three different people from the *Sports* dataset and performed an exhaustive pairwise comparison among the 8 photos. Hence, there were $\binom{8}{2}$ = 28 possible pairwise comparisons. For the $\mathcal{F}$ experiment, we first selected 6 photos of 3 family members from the *Family* dataset. For each family member, we chose 3 photos when the person was an infant and 3 photos when the person was a child. We then chose 28 random pairs of images to compare where 14 pairs were matching and 14 were non-matching. We chose the same numbers of matching and non-matching pairs in order to prevent a worker that only answered No (or Yes) from obtaining a high accuracy. We used an identical selection method for the $\mathcal{M}$ and $\mathcal{C}$ experiments. For the $\mathcal{G}_{11}$ experiment, we selected 14 matching and 14 non-matching pairs from photos of 11 gymnasts. Finally for the $\mathcal{D}$ experiment, we first selected from the *Dots* dataset 5 images containing 100 dots each, 5 images containing 150 dots each, and so on until we collected a total of 50 images. Then among those images, we chose 14 random matching and 14 random non-matching image pairs.

For the 28 record pairs in each experiment, we shuffled them and replicated them 9 times each to create a total of 252 questions. We then started filling HITs with the questions. We perform these comparisons for both the Y/N and $Y/N/M$ interfaces. Since we payed a worker 2 cents per HIT, we payed around 2 dollars to the workers per experiment. When posting the HITs, we divided the HITs into 4 batches that were posted at least 1 hour apart. We spread out the workload in order to prevent the same workers from solving too many HITs and getting used to the comparisons.

We restricted the worker demographics for all the experiments in Table II to be people living in the US having an approval

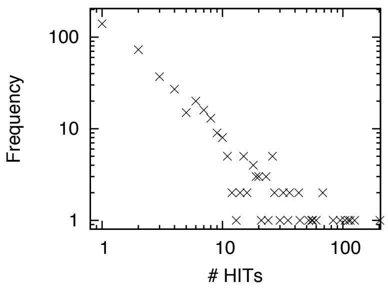


Fig. 5. Number of HITs per worker

rate over 90%. The approval rate restriction requires the workers to have a good record of correctly solving other tasks more than 90% of the time. Figure 5 shows (on a log-log scale) how many HITs workers performed for all the experiments in Table II. A total of 414 unique workers performed 8 HITs on average. Most of the workers performed few HITs (e.g., 140 workers performed 1 HIT) while fewer workers performed many HITs (e.g., there was one worker who performed 205 HITs).

Accuracy Measures: We used two measures to quantify the comparison results. The *overall accuracy* is identical to our accuracy definition in Section II and divides the number of correct answers by the total number of questions. The *majority accuracy* first takes the majority answer for each question. Since each question is repeated 9 times, we consider the human answer to be Yes if there are at least 5 Yes answers and No otherwise. The measure then divides the number of correctly classified image pairs by the total number of image pairs compared. When evaluating results using the $Y/N/M$ interface, a question may have fewer than 9 Yes or No answers. If the numbers of Yes and No answers are the same, we flip a coin to decide if the majority answer is Yes or No.

B. Accuracy Results

Table III shows the comparison accuracy results for the Y/N and $Y/N/M$ interfaces. (For each experiment, the best accuracy is highlighted in bold.) We answer the following questions: First, how does the Maybe option improve accuracy without majority voting? Second, how does majority voting improve accuracy without the Maybe option? Third, does majority voting improve accuracy in the presence of the Maybe option? We then study the variability of the Table III accuracies and how the average HIT completion time relates to accuracy.

1) Maybe Results: Table III shows how the Maybe option improves the accuracy of the Y/N interface. For example, experiment $\mathcal{G}$ has an improvement from 0.849 to 0.910. Figure 6 gives a more detailed view of experiment $\mathcal{G}$ by showing how individual workers performed in the $\mathcal{G}$ experiment. The x axis represents all the workers while the y axis shows the accuracies of the workers. For the Y/N plot, the accuracy of a worker is computed as the number of correct answers divided by the total number of her answers. For the $Y/N/M$ plot, the accuracy of a worker is the number of correct answers divided by

Exp.	Y/N **Interface**		$Y/N/M$ **Interface**	
	Overall	**Majority**	**Overall**	**Majority**
People				
$\mathcal{G}$	0.849	0.893	0.910	**0.964**
$\mathcal{G}_{11}$	0.881	**0.964**	0.890	0.893
$\mathcal{S}$	0.688	**0.833**	0.689	0.798
$\mathcal{B}$	0.892	**0.929**	0.919	0.917
$\mathcal{F}$	0.665	0.702	0.700	**0.714**
$\mathcal{M}$	0.861	**0.958**	0.870	0.923
Objects				
$\mathcal{C}$	0.691	0.679	0.706	**0.714**
Abstract				
$\mathcal{D}$	0.806	**0.929**	0.809	0.821

TABLE III

ACCURACY RESULTS FOR VARIOUS APPLICATIONS

the number of her non-Maybe answers. The numbers in parentheses indicate the number of workers evaluated and thus are equal to the numbers of points in the plots. We sort the workers by their accuracies on a normalized x axis. The worker with the lowest accuracy has an x value of 0 while the worker with the highest accuracy has an x value of 1. By comparing the two plots, we observe that the worker accuracies of the $Y/N/M$ interface are mostly higher than the Y/N accuracies, which means that the Maybe option was indeed filtering out answers more likely to be incorrect.

While using the Maybe option always improves the comparison accuracies of experiments, the amount of improvement is sometimes subtle. For example, experiment $\mathcal{D}$ only has an improvement of 0.809-0.806 = 0.003. In this case, we believe the tasks were easy to the extent that guessing a Yes or No answer (when the Maybe option was not offered) produced correct results.

Another interesting question when using the Maybe option is how frequently workers choose Maybe for different experiments. Figure 7 shows the overall accuracy of using the $Y/N/M$ interface versus the portion of Maybe answers for all the experiments in Table II. For experiments with low accuracies (which implies that the tasks are more difficult), the portion of Maybe answers is generally high. For example, experiment $\mathcal{F}$ has an accuracy of 0.7 where 27% of the answers are Maybe. The experiments with high accuracies tend to have low portions of Maybe answers. For example, experiment $\mathcal{G}_{11}$ has an accuracy of 0.89 where only 6% of the answers are Maybe. The most noticeable outlier is experiment $\mathcal{C}$, which has an accuracy of 0.706, but

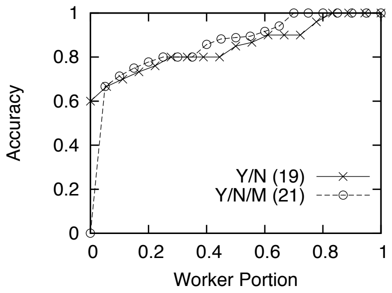


Fig. 6. Worker accuracy distribution ($\mathcal{G}$)

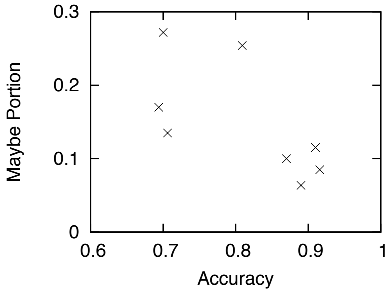


Fig. 7. Accuracy impact on Maybe frequency

a small Maybe portion of 13%. Here, some of the camera lens comparisons were subtle and may have given the workers a false impression that they were doing a good job. This result illustrates how workers may have different levels of confidence when comparing products instead of people.

2) Majority Results: Figure 8 shows how much majority voting improves the Y/N overall accuracies in Table III. Each experiment is represented as an 'x' point where the x value is the overall accuracy using the Y/N interface, and the y value is the majority accuracy minus the overall accuracy using the Y/N interface. For example, experiment $\mathcal{M}$ has an accuracy improvement from 0.861 to 0.958, so we draw the point (0.861, 0.958-0.861) = (0.861, 0.097). Being above the $y = 0$ line thus means that majority voting improved the accuracy. We observe that majority voting mostly improves the accuracy, but may also decrease the accuracy. For example, suppose that there are two questions q_1 and q_2 that are replicated 9 times each by the Replicate module. Say that q_1 is easy, and all the 9 answers by workers are correct. On the other hand, let us assume that q_2 is difficult, and only 4 out of 9 of the answers are correct. If we treat all the replicated questions as different questions, the overall accuracy is $\frac{9+4}{9+9} = 0.72$. However, if we use majority voting, then the accuracy for the two questions is only $\frac{1+0}{1+1} = 0.5$.

We compare the 'x' points with the expected accuracy improvement when all the questions have the same and independent probability p of being answered correctly. Since we replicate each question 9 times, the expected accuracy improvement is $\sum_{i=5\ldots9} p^i(1-p)^{9-i}$ (see Section II-B for the general formula) minus p, which is the overall accuracy. This formula is plotted with the label "Uniform (0% Maybe)". We observe that all the 'x' points are below the curve. The difference from the curve

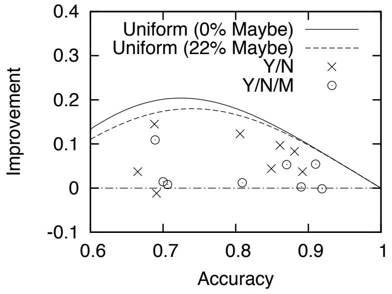


Fig. 8. Majority accuracy improvement

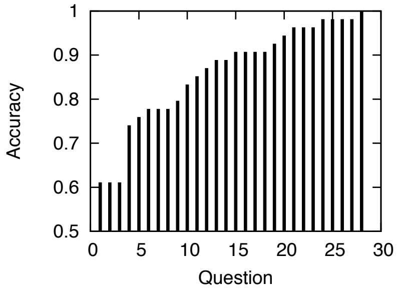


Fig. 9. Accuracy per question (movie stars)

illustrates how much the actual error probabilities deviate from the ideal model (independent errors by workers). Of course, in any given experiment, one can expect some deviation, but the larger deviations from the ideal model are less likely.

The main reason for the lower-than-ideal accuracy improvements is the uneven difficulties of comparisons. For example, Figure 9 shows the varying comparison accuracies for the questions in experiment $\mathcal{M}$. Even within the same experiment, the questions clearly have different levels of difficulty. As a result, the questions are no longer independent because incorrect answers tend to concentrate on the difficult questions.

3) Majority with Maybe Results: Using majority voting on top of the $Y/N/M$ interface mostly increases the accuracy, but again the improvements can be subtle or even negative (e.g., experiment $\mathcal{B}$). In Figure 8, each experiment using the $Y/N/M$ interface is represented as an 'o' point where the x value is the overall accuracy using the $Y/N/M$ interface, and the y value is the majority accuracy minus the overall accuracy using the $Y/N/M$ interface. For example, experiment $\mathcal{M}$ has an accuracy improvement from 0.87 to 0.923, so we draw the point (0.87, 0.923-0.87) = (0.87, 0.053). The improvements are relatively worse than those for the Y/N interface because majority voting was not as effective with fewer non-Maybe answers.

We would like to compare the 'o' points with a uniform probability scenario. However, the "Uniform (0% Maybe)" plot assumes that there are no Maybe answers, which is not the case when using the $Y/N/M$ interface. Hence, we consider a more relevant scenario where 2 out of 9 repeated questions are assumed to be Maybe answers. The other questions are answered correctly with probability p. Since we now take the majority result of 7 non-Maybe answers per question, the expected accuracy

Exp.	Y/N **Interface**		$Y/N/M$ **Interface**	
	Overall	**Majority**	**Overall**	**Majority**
$\mathcal{S}$	0.662–0.713	0.800–0.867	0.659–0.720	0.755–0.828
$\mathcal{B}$	0.881–0.904	0.904–0.953	0.903–0.935	0.887–0.947
$\mathcal{F}$	0.644–0.685	0.682–0.723	0.675–0.726	0.701–0.715
$\mathcal{M}$	0.835–0.887	0.938–0.978	0.844–0.896	0.907–0.962

TABLE IV

CONFIDENCE INTERVALS (80%)

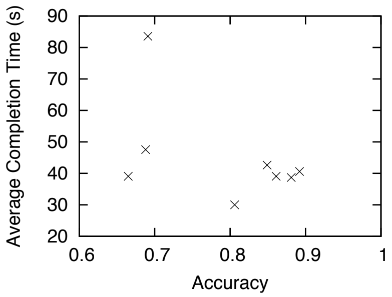


Fig. 10. Accuracy impact on HIT completion time

improvement is $\sum_{i=4\ldots7} p^i(1-p)^{7-i} - p$. In Figure 8, we plot this formula with the label "Uniform (22% Maybe)". We observe that all the 'o' points are below the curve again because of the uneven difficulties of questions.

4) Variability: To study the variability in the Table III results, we repeated the $\mathcal{S}$, $\mathcal{B}$, $\mathcal{F}$, and $\mathcal{M}$ experiments 6 times each. The accuracy results in Table III for the four experiments are the averaged results. Table IV shows the 80% confidence intervals of the accuracies. We observe that Maybe option accuracy improvements on the Y/N interface are not significant for any experiment. Next, the majority voting improvements on the Y/N interface are significant for the $\mathcal{S}$, $\mathcal{B}$, and $\mathcal{M}$ experiments. Finally, the majority voting improvements on the $Y/N/M$ interface are significant for the $\mathcal{S}$ and $\mathcal{M}$ experiments. Although we can further reduce the confidence intervals by repeating the experiments, we can already see that using majority voting on the Y/N interface results in the most significant improvements.

5) HIT Completion Time: Figure 10 shows how the overall accuracy of an experiment using the Y/N interface relates to the average HIT completion time. We observe that the completion times vary significantly and do not show any particular trend against different accuracies. Even for repetitions of the same experiment (which were performed in Section IV-B4), the completion times were very different. For example, the average HIT completion times for the 6 repetitions of experiment $\mathcal{B}$ ranged from 24 to 66 seconds.

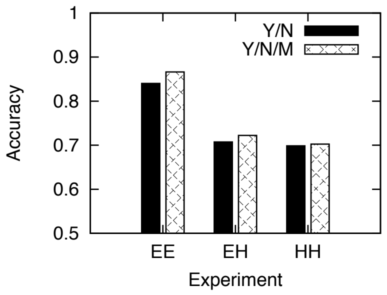


Fig. 11. Image clarity impact on accuracy

C. Task Difficulty

The difficulty of a comparison can impact the accuracy. Here we explore two possible subjective notions of difficulty, one based on image "clarity" and one based on age differences. In the following sub-sections, we explore the two definitions of difficulty and see how they impact the comparison accuracy.

1) Image Clarity: We study how the image clarity impacts the comparison accuracy. We collected all the person and camera lens photos in the experiments of Table II and manually annotated each photo as "easy" (E) or "hard" (H) depending on how hard it was to identify the person or object. For example in Figure 4, photos 1, 2, and 7 were considered hard because the postures (heading upside down or sideways) of the people made it difficult to identify their faces. Out of a total of 124 person and camera photos, 27 were tagged as hard, and the remaining photos were considered easy. (Since evaluating difficulty is subjective, our method is not the only way to categorize photos.) Figure 11 shows the overall accuracy for comparing easy photos with each other (denoted as EE), comparing easy photos with hard photos (denoted as EH), and comparing hard photos with each other (denoted as HH). As the difficulty of the comparison increases, the accuracy decreases. The individual difficulty of images can thus be useful in explaining the accuracies we obtain.

2) Age Difference: We study how the pairwise age difference between photos can affect the comparison accuracy. We use the same image pairs in the experiments $\mathcal{M}$ and $\mathcal{F}$ and derive the accuracies using the Y/N or $Y/N/M$ interfaces. For each dataset, we extract the photo pairs that are both about the younger versions of people (denoted as YY) and measure the overall accuracies. Next, we measure the accuracy for photo pairs where one photo is young and the other is old (denoted as YO). For example, images 7 and 8 in Figure 4 form a YO pair for the $\mathcal{F}$ experiment. Finally, we repeat the same computation for photo pairs containing the older versions of people (denoted as OO). Figure 12 shows that, when comparing movie stars (experiment $\mathcal{M}$), the YO comparisons have the worst overall accuracy results because the age difference makes it hard to identify the same person. For the $\mathcal{F}$ experiment, however, the YY comparison accuracies are even lower than the YO accuracies because the young photos showed people when they were infants. In general, comparing infants is harder than comparing children.

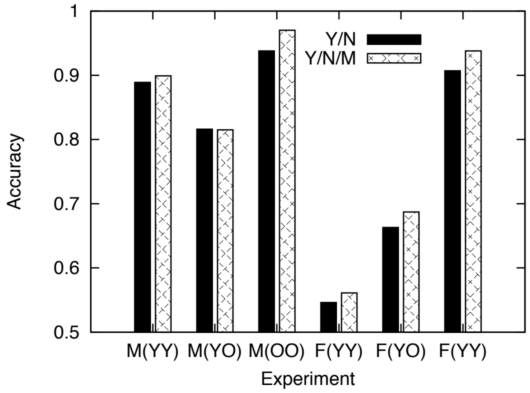


Fig. 12. Age difference impact on accuracy

Exp.	Y/N **Interface**		$Y/N/M$ **Interface**	
	Overall	**Majority**	**Overall**	**Majority**
$\mathcal{G}$	0.849	0.893	0.910	0.964
$\mathcal{G}_I$	0.583	0.607	0.565	0.643

TABLE V

DEMOGRAPHICS IMPACT ON ACCURACY

Moreover, since the infants were relatives, they resembled each other, making the comparisons even harder. Hence, just like the image clarity, the pairwise time information of images can also be useful in explaining why we obtain certain accuracies.

D. Worker Ability

In addition to the difficulty of the HITs themselves, the worker ability may also influence the comparison accuracy. We study two skill factors: the demographics of workers and the experience they obtain while solving HITs.

1) Demographics: We vary the demographics of the workers to see how they influence the comparison accuracy. Compared to the default demographics (US workers with more than 90% approval rate), we now restrict the workers to live in India. We still require the workers to have an approval rate of more than 90%. We then perform the same pairwise comparisons as $\mathcal{G}$ and call the new experiment $\mathcal{G}_I$. Table V shows that the Indian workers indeed have more trouble distinguishing the gymnasts than the US workers. Since the athletes were non-Indians, we suspect that comparing people of different ethnicity was the major reason for the poor performance. On average, the Indian workers were spending 192 seconds per HIT while a US worker spent 43 seconds for experiment $\mathcal{G}$. Hence, the Indian workers were performing poorly despite the more time they spent. Moreover, providing the Maybe option did not seem to help either.

Exp.	Y/N **Interface**		$Y/N/M$ **Interface**	
	Overall	**Majority**	**Overall**	**Majority**
$\mathcal{C}$	0.691	0.679	0.706	0.714
$\mathcal{C}_{2,10}$	0.635	0.643	0.679	0.857
$\mathcal{C}_{4,10}$	0.774	0.857	0.782	0.750

TABLE VI

EXPERIENCE AND WAGE

2) Experience: More worker experience can help improve the ER quality, given that the worker is paid proportionally to the increased amount of work. We replicate the camera lens comparisons in experiment $\mathcal{C}$, but we now ask 10 questions per HIT instead of 5 questions. In addition, we pay 4 cents instead of 2 cents because a worker is performing twice as much work as before. We call this new experiment $\mathcal{C}_{4,10}$. Table VI shows that $\mathcal{C}_{4,10}$ clearly improves in accuracy compared to $\mathcal{C}$ in all columns. We were also interested if we could simply ask 10 questions without raising the wage and have similar results. We thus performed an experiment called $\mathcal{C}_{2,10}$, which is identical to $\mathcal{C}_{4,10}$ except that the wage per HIT is 2 cents. As a result, the accuracies are no better than $\mathcal{C}$. Hence, more experience can lead to higher accuracy, but only if the workers are payed in proportion to the increased amount of work they perform.

E. Other Interfaces

We now explore two extensions of the Y/N and $Y/N/M$ interfaces. The first extension asks for answers within a range (instead of a Yes or No answer) while the second extension compares clusters of images (instead of two images).

1) Answers within a Range: We first explore whether asking for an answer within a range of values can help improve the comparison accuracy. (The problem of setting the number of choices in a questionnaire has been well studied in marketing research [13].) We perform an experiment (called $\mathcal{F}_R$) that compares the same photo pairs as $\mathcal{F}$, but now uses a range interface that requires an integer within the range [1–5] instead of a Yes, No, or Maybe answer. In the instructions of each HIT, we added the explanation that option 1 means the images do not match, option 5 means that the images do match, option 3 means that the worker is uncertain, and option 4 means that the worker is leaning towards a match. Hence, we are providing more options for the workers to express uncertainty.

One way to compare the results of $\mathcal{F}_R$ with $\mathcal{F}$ is to convert the integers within the range [1–5] into Yes, No, or Maybe answers. Specifically, a range answer larger than 3 is considered as Yes, an answer smaller than 3 a No, and an answer equal to 3 a Maybe. We can then compute the overall accuracy of the converted result. Table VII shows that $\mathcal{F}_R$ has an overall

Exp.	Overall Accuracy	Weighted Accuracy
$\mathcal{F}$	0.700	0.640
$\mathcal{F}_5$	0.696	0.610
$\mathcal{F}_7$	0.711	0.677

TABLE VII

RANGE INTERFACE RESULTS

accuracy of 0.696, which is very similar to the 0.7 overall accuracy of $\mathcal{F}$ using the $Y/N/M$ interface. This result suggests that, if we are only interested in a Yes or No answer in the end, then the $Y/N/M$ and range interfaces produce similar accuracies.

Another possible comparison is to convert the Yes, No, or Maybe answers of $\mathcal{F}$ into integers within the range [1–5] and compute a weighted accuracy. Here, we can consider a Yes answer as a 5, a No answer as a 1, and a Maybe as a 3. To give different weights for the integers in the range, we can define the accuracy of each worker comparison result as 1 minus the absolute difference between the worker's answer and the correct answer divided by 4. For example, if two photos are the same and a worker answers 4, then the comparison accuracy is $1 - \frac{|4-5|}{4} = 0.75$. The final accuracy is then the sum of the individual comparison accuracies divided by the number of comparisons. Using this measure, $\mathcal{F}$ and $\mathcal{F}_R$ have the weighted accuracies 0.64 and 0.61, respectively. Compared to the overall accuracy results, the two accuracies have a larger difference, which suggests that the weighted accuracy is more sensitive to the uncertainty of workers.

Would increasing the number of options further improve the range interface? Experiment $\mathcal{F}_7$ uses a range interface that now requires an integer answer within [1–7]. We again compute the overall and weighted accuracies using measures analogous to those for $\mathcal{F}_5$. As a result, Figure VII shows that $\mathcal{F}_7$'s overall accuracy 0.711 is slightly higher, but still similar to the overall accuracies of $\mathcal{F}$ and $\mathcal{F}_5$. On the other hand, the weighted accuracy 0.677 is significantly higher than both $\mathcal{F}$ and $\mathcal{F}_5$. Again, we observe that the weighted accuracy is more sensitive to the uncertainty of workers than the overall accuracy. However, the sensitivity of the weighted accuracy does not seem to be greater than experiment $\mathcal{F}_5$. We thus conclude that using more than 5 options does not necessarily help capture the worker uncertainty better.

2) Comparing Clusters: When comparing two possible entities, it can be useful to view more images that already match with either of the two entities. For example, suppose a worker is comparing two images a and b of the same baseball player. Say that in image a, the worker can observe the name of the player, but not the jersey number. For image b, say the worker can observe the jersey number of the player, but not the name. Hence the worker cannot tell the images refer to the same entity. Instead, say that on the left we show two images, a and c that are known to refer to the same athlete, and ask the worker if this athlete matches the one in the right image b. If image c shows the athlete's number on her jersey, then the worker may

Exp.	Y/N **Interface**		$Y/N/M$ **Interface**	
	Overall	**Majority**	**Overall**	**Majority**
$\mathcal{F}$	0.665	0.702	0.700	0.708
$\mathcal{F}_{2-1}$	0.798	0.821	0.813	0.821
$\mathcal{F}_{2-2}$	0.837	0.929	0.866	0.964

TABLE VIII

CLUSTER INTERFACE RESULTS

make the identification.

We extend the Y/N and $Y/N/M$ interfaces by adding one image per question. For each pair of images a and b compared, we display them as before, but also add an image c that we know matches with a. Image c is randomly chosen from all the images (other than a and b) that match with a. The images a and c are vertically stacked upon each other forming a cluster next to b. Table VIII compares the experiment $\mathcal{F}$ with an experiment (called $\mathcal{F}_{2-1}$) that compares the exact same pairs of photos as $\mathcal{F}$, but also extends each question by one image as described above. As a result, $\mathcal{F}_{2-1}$ clearly improves the overall accuracy of $\mathcal{F}$ using the Y/N interface. In addition, we observe the same accuracy improvements when using the Maybe option or majority voting.

We further extend the two interfaces by adding yet another image per question. For each pair of images a and b, we first add an image c as before. In addition, we choose another random image d from all the images (other than a, b, and c) that match with b. We now vertically stack a and c on the left side of the question and stack b and d on the right side of the question. Table VIII shows the new experiment $\mathcal{F}_{2-2}$ that uses this extension. Compared to $\mathcal{F}_{2-1}$, the overall accuracy again improves because the additional images give more hints. We also observe the additional improvements in accuracy when using the Maybe option or majority voting.

V. WORKER ANALYSIS

A useful analysis of workers is to categorize them as good and bad based on how they perform. A good worker would answer questions with high accuracy while a bad worker would not. The main benefit is that we can potentially improve ER accuracy by assigning future tasks to the good workers. We discuss two approaches – threshold-based and model training – for the categorization.

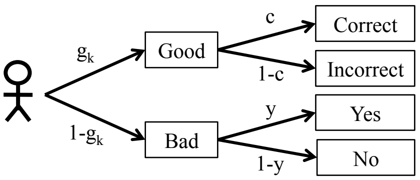


Fig. 13. Worker model for Y/N interface

A. Accuracy Threshold

One way to label the workers is to compare their comparison accuracies with a threshold. Consider the scenario where we test the worker by showing her tasks for which we know the answer in advance (a gold standard). These tasks may be shown in a testing phase, or may be interspersed with real tasks for which we do not have an answer. We then evaluate a worker based on her accuracy, i.e., on her response to the gold standard tasks. If the accuracy is larger than some threshold T, then we categorize the worker as good. Otherwise, the worker is bad. While the categorization method is straightforward, the challenge is to define the right threshold T. For example, recall that Figure 6 in Section IV-B shows the individual worker accuracies for the $\mathcal{G}$ experiment. Here we could consider any accuracy over $T = 0.5$ as high and thus categorize most of the workers as good. Or since the two curves seems to taper off from an accuracy of 0.8, we could set the threshold as $T = 0.8$. Hence, setting the threshold can be subjective.

B. Model Training

Another way to categorize the workers is to train a worker model that captures the good and bad behaviors of all the workers without using a threshold. A worker who fits the good worker model is defined as good while a worker who fits the bad worker model is defined as bad. We now propose worker models for the Y/N and $Y/N/M$ interfaces.

Y/N Interface: Figure 13 shows our worker model for the Y/N interface. Each worker $w_k \in W$ is categorized as good if $g_k = 1$ and bad if $g_k = 0$. If a worker is good, then she is correct with probability c and incorrect with probability $1 - c$. If the worker is bad, she answers Yes (independent of what is the correct answer) with probability y and No with probability $1 - y$. If the good workers have high accuracy, then c will be high as well. Notice that there is a g_k binary variable for each worker w_k while the c and y probabilities are common to all workers.

We can train the worker model using a set of record comparison results S by workers. Each comparison result can be represented as an observation (w_k, v, a) where w_k identifies the worker, v is a Yes or No answer by w_k, and a is the correct Yes or No answer. Let us denote the set of comparison results for a worker w_k as I_k. We would like to compute the likelihood of $S = \bigcup_{w_k \in W} I_k$ from the worker model by computing the probability product of the individual (and independent) likelihoods over all the observations. In order to ensure that c and y are between 0 and 1, we define them as $\frac{e^{c'}}{Z_1}$ and $\frac{e^{y'}}{Z_2}$, respectively where Z_1 and Z_2 are normalization constants defined as $e^{c'}+1$ and $e^{y'}+1$, respectively. Suppose we denote the probability of observing (w_k, v, a) as $p(w_k$, v, $a)$. The likelihood $\mathcal{L}(S)$ is then

$$\begin{aligned}&\prod_{w_k \in W} \prod_{i \in I_k} p(w_k, v, a) = \\ &\prod_{w_k \in W} \prod_{i \in I_k} (g_k \times (1\{v_i = a_i\} \times \frac{e^{c'}}{Z_1} + 1\{v_i \neq a_i\} \times \frac{1}{Z_1}) \\ &\qquad + (1 - g_k) \times (1\{v_i = \text{Yes}\} \times \frac{e^{y'}}{Z_2} + 1\{v_i = \text{No}\} \times \frac{1}{Z_2}))\end{aligned}$$

where the indicator function $1\{b\}$ takes on a value of 1 if its argument b is true and 0 otherwise.

We can maximize $\mathcal{L}(S)$ using coordinate ascent [14] on the log likelihood $\log \mathcal{L}(S)$ shown as follows. (We add the penalty function $-\lambda 1\{g_k > 0\}$ for the purpose of breaking ties between equally good solutions.)

$$\begin{aligned}&\sum_{k \in W} (\sum_{i \in I_k} \log(g_k \times (1\{v_i = a_i\} \times \frac{e^{c'}}{Z_1} + 1\{v_i \neq a_i\} \times \frac{1}{Z_1}) \\ &\qquad + (1 - g_k) \times (1\{v_i = \text{Yes}\} \times \frac{e^{y'}}{Z_2} + 1\{v_i = \text{No}\} \times \frac{1}{Z_2})) \\ &\qquad - \lambda 1\{g_k > 0\})\end{aligned}$$

The coordinate ascent is done in two alternating steps. First, we fix the g_k values and use gradient ascent to derive the c' and y' values that maximize $\log \mathcal{L}(S)$. Next, we fix c' and y' and derive the optimal g_k values that maximize $\log \mathcal{L}(S)$. We repeat the two steps until $\log \mathcal{L}(S)$ does not change significantly. In general, using coordinate ascent does not guarantee a global optimum, but does guarantee a local optimum. We can keep on improving $\log \mathcal{L}(S)$ by repeating coordinate ascent using randomly initialized parameters until $\log \mathcal{L}(S)$ does not increase significantly.

To illustrate the worker model training, suppose that there are two workers w_1 and w_2. Say that w_1 has correctly answered two questions by giving the answers Yes and No, and that worker w_2 answers one question incorrectly by answering No. We thus have three worker responses (w_1, Yes, Yes), (w_1, No, No), and (w_2, No, Yes). Suppose that we initially consider both workers to be bad, i.e., $g_1 = g_2 = 0$. Using 5 trials of coordinate ascent, we arrive at a likelihood $\mathcal{L}(S)$ of 0.996 where $g_1 = 1$, $g_2 = 0$, $c = 0.998$, and $y = 0.005$ (see Section V-C for more training results). That is, w_1 is categorized as a good worker that is almost always correct and w_2 a bad worker that almost always answers No, which indeed reflects the three responses.

Y/N/M Interface: We can extend the worker model of the Y/N interface for the $Y/N/M$ interface as illustrated in Figure 14. For each type of worker (i.e., good and bad), we add a Maybe branch. Again, there is a probability g_k for each worker w_k for being good or bad. For all the workers, there are now four probabilities – c, i, y, and n – that need to be trained. The log likelihood can again be maximized using repeated trials of coordinate ascent on randomized parameters.

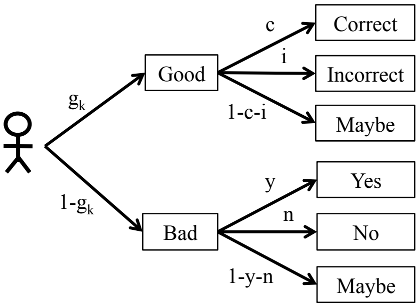


Fig. 14. Worker model for $Y/N/M$ interface

C. Model Evaluation

We now use the worker models to categorize workers. For each experiment in Table II, we performed 20 trials of coordinate ascent on randomly-initialized parameters. Table IX shows the training results for all the experiments. The column "Ratio" contains the fraction of good workers, i.e., $\frac{\sum_{k \in W} g_k}{|W|}$. For example, if 6 out of 10 workers are labeled as good, then the good worker ratio is 0.6. For the Y/N interface results, we show the values of the c and y parameters. For example, the good workers in experiment $\mathcal{M}$ using the Y/N interface are correct with $c = 0.93$ probability while the bad workers answer Yes with $y = 0.31$ probability. For the experiments using the $Y/N/M$ interface, we show the good worker ratio and the values $\bar{c} = \frac{c}{c+i}$ and $\bar{y} = \frac{y}{y+n}$ where $\bar{c}$ is the fraction of correct answers among all non-Maybe answers by good workers, and $\bar{y}$ is the fraction of Yes answers among all non-Maybe answers by bad workers. For example, the trained parameters of experiment $\mathcal{M}$ using the $Y/N/M$ interface are $c = 0.87$, $i = 0.08$, $y = 0.23$, and $n = 0.49$. In this case, $\bar{c} = \frac{0.87}{0.87+0.08} = 0.92$, and $\bar{y} = \frac{0.23}{0.23+0.49} = 0.32$.

We observe that the $Y/N/M$ worker model tends to produce lower ratios of good workers that have higher correctness probabilities compared to the Y/N interface worker model. The results show that good workers using the $Y/N/M$ interface improve their accuracies by utilizing the Maybe option. Another observation is that the y values of the Y/N model are mostly higher than the $\bar{y}$ values of the $Y/N/M$ model. One explanation is that the Y/N workers who were uncertain about their answers chose Yes as a default instead of No mainly because the Yes radio button was always placed on top of the No button for each question.

VI. Related Work

Entity Resolution has been studied under various names including record linkage, merge/purge, deduplication, reference reconciliation, object identification, and others (see [5], [19] for recent surveys). Many ER algorithms can benefit from pairwise record comparisons done by humans. For example, a large class of ER algorithms first identify candidate matching pairs of records, and then determine the final pairs of matching records [9], which can be done by humans.

Exp.	Y/N **Interface**			$Y/N/M$ **Interface**		
	Ratio	c	y	Ratio	$\bar{c}$	$\bar{y}$
$\mathcal{G}$	0.79	0.88	0.17	0.48	0.95	0.05
$\mathcal{G}_{11}$	0.75	0.93	0.70	0.56	0.94	0.52
$\mathcal{G}_I$	0.64	0.71	0.85	0.64	0.78	0.88
$\mathcal{S}$	0.72	0.78	0.99	0.60	0.82	0.90
$\mathcal{B}$	0.92	0.91	0.79	0.87	0.96	0.44
$\mathcal{F}$	0.68	0.74	0.40	0.66	0.69	0.33
$\mathcal{M}$	0.63	0.93	0.31	0.61	0.92	0.32
$\mathcal{C}$	0.50	0.78	0.18	0.85	0.77	0.11
$\mathcal{D}$	0.75	0.82	0.25	0.40	0.88	0.15

TABLE IX

GOOD WORKER RATIOS AND PROBABILITIES

Human learning techniques [12] have recently been proposed for ER. Reference [8] performs unsupervised learning based on humans clustering a subset of records and then applies the trained clustering algorithm to the entire dataset. Reference [2] has recently used active learning techniques for ER where the idea is to only learn the necessary information for training the ER algorithm. Recently, human resolution techniques for ER have been proposed as well. Reference [20] proposes a human resolution system where authors can claim their own publications. ZenCrowd [4] uses the crowd to figure out which entities in web pages refer to the same URI. In comparison, our techniques focus on finding the best interface for an ER algorithm to interact with the crowd.

Several interfaces have been used for crowd ER. CrowdDB [6] uses a Y/N interface for comparing records. Qurk [15] uses a mapping interface where multiple records on one side of the interface are mapped to records on the other side. CrowdER [17] uses two interfaces: the Y/N interface and a clustering interface where workers can group the same records. In comparison, our work performs a detailed comparison of various pairwise interfaces and identifies the major factors (including non-interface ones) that influence the comparison accuracy.

Worker analysis has usually been done on the entire set of workers for platforms like Amazon Mechanical Turk [11], [16]. We instead focus on a relatively narrow class of workers performing pairwise image comparisons. While some of our results conform to the general results in the literature (e.g., workers perform better with more wage), we have also identified behaviors specific to image comparisons as well (e.g., how uncertain workers are when comparing products instead of people).

Interfaces have also been studied in other areas as well. In HCI, SUPPLE++ [7] has been proposed as a tool for generating user interfaces adapted to users' motor and vision capabilities. In comparison, our work focuses on maximizing the comparison accuracy of records given a fixed budget. Marketing research [13] has heavily studied various design choices for questionnaires. While our work only considers a subset of these design choices that are relevant to pairwise image comparisons, other designs can be used in our interfaces as well.

VII. Conclusion

We have studied the problem of enhancing ER using crowdsourcing with a focus on the interface between workers and the ER algorithm. In particular, we have extensively studied two pairwise interfaces: Y/N and $Y/N/M$. We have evaluated the two interfaces using Amazon Mechanical Turk with real and synthetic datasets. Our results suggest that using the Maybe option or taking the majority of answers improves the comparison accuracy, but sometimes only by a subtle amount. In addition, the difficulty of the tasks and the worker ability can significantly influence the comparison accuracy. We have observed that an interface asking for an answer within a range does not necessarily collect more information than the $Y/N/M$ interface if we only require a Yes or No answer in the end. Finally, comparing clusters of matching records can provide more information to the workers and improve accuracy. We also proposed two methods for worker analysis: one that uses a threshold and another that trains a worker model without a threshold. By identifying the good workers, we can potentially improve the ER accuracy by assigning future tasks to the good workers.

We believe our work is the first extensive study on pairwise comparison interfaces for ER and that there are many future directions for research. First, we would like to extend our study to interfaces that compare more than two records at a time, requiring workers to either map or cluster records. Second, we can combine our worker interaction study with any ER algorithm that asks questions to humans. It would be interesting to see how the Crowd ER algorithm in reference [18] will work with our Y/N and $Y/N/M$ interfaces. One key assumption of the Crowd ER algorithm is that workers give perfect answers (i.e., the accuracy is always 1). However, we have observed that the worker accuracy can be influenced by various factors, so while our techniques can help improve the accuracy, it is not realistic to assume the workers will always be correct. How to incorporate the imperfect answers back into the ER process is an interesting challenge.

References

[1] Amazon mechanical turk. https://www.mturk.com.

[2] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *SIGMOD*, pages 783–794, 2010.

[3] Crowdflower. http://crowdflower.com.

[4] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, New York, NY, USA, 2012.

[5] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[6] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD*, pages 61–72, 2011.

[7] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *CHI*, pages 1257–1266, 2008.

[8] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, pages 558–566, 2011.

[9] O. Hassanzadeh, F. Chiang, R. J. Miller, and H. C. Lee. Framework for evaluating clustering algorithms in duplicate detection. *PVLDB*, 2(1):1282–1293, 2009.

[10] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *SIGMOD*, pages 127–138, 1995.

[11] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *ACM Crossroads*, 17(2):16–21, 2010.

[12] E. Law and L. von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[13] P. Lietz. Research into questionnaire design. *International Journal of Market Research*, 52(2):249–272, 2010.

[14] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[15] A. Marcus, E. Wu, D. RKarger, S. Madden, and R. Miller. Human-powered sorts and joins. *PVLDB*, 5(1):13–24, 2011.

[16] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

[17] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[18] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. Technical report, Stanford University, available at http://ilpubs.stanford.edu:8090/1047/.

[19] W. Winkler. Overview of record linkage and current research directions. Technical report, Statistical Research Division, U.S. Bureau of the Census, Washington, DC, 2006.

[20] Y. Yang, P. Singh, J. Yao, C. man Au Yeung, A. Zareian, X. Wang, Z. Cai, M. Salvadores, N. Gibbins, W. Hall, and N. Shadbolt. Distributed human computation framework for linked data co-reference resolution. In *ESWC (1)*, pages 32–46, 2011.