Proceedings, The Fourth AAAI Conference on
Human Computation and Crowdsourcing

MicroTalk: Using Argumentation to Improve Crowdsourcing Accuracy

Ryan Drapeau, Lydia B. Chilton, Jonathan Bragg, Daniel S. Weld
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
{drapeau, hmslydia, jbragg, weld}@cs.washington.edu

Abstract

Crowd workers are human and thus sometimes make mistakes. In order to ensure the highest quality output, requesters often issue redundant jobs with gold test questions and sophisticated aggregation mechanisms based on expectation maximization (EM). While these methods yield accurate results in many cases, they fail on extremely difficult problems with local minima, such as situations where the majority of workers get the answer wrong. Indeed, this has caused some researchers to conclude that on some tasks crowdsourcing can *never* achieve high accuracies, no matter how many workers are involved.

This paper presents a new quality-control workflow, called MicroTalk, that requires some workers to *Justify* their reasoning and asks others to *Reconsider* their decisions after reading counter-arguments from workers with opposing views. Experiments on a challenging NLP annotation task with workers from Amazon Mechanical Turk show that (1) argumentation improves the accuracy of individual workers by 20%, (2) restricting consideration to workers with complex explanations improves accuracy even more, and (3) our complete MicroTalk aggregation workflow produces much higher accuracy than simpler voting approaches for a range of budgets.

Introduction

Crowdsourcing, the outsourcing of tasks to a crowd of unknown people ("workers") in an open call, is rapidly rising in popularity. It is already being heavily used by numerous employers ("requesters") for tasks ranging from audio transcription to NLP data annotation. However, ensuring output quality remains a key challenge, because of the high variability in worker abilities. The use of "gold-standard" screening questions can help, but even the best workers make mistakes. To achieve the highest quality results, therefore, most requesters issue their tasks to multiple, independent workers and aggregate the results. A variety of techniques have been tried, but most employ some sort of weighted majority vote or expectation maximization (Dawid and Skene 1979). While these methods yield accurate results in many cases, they often converge to local maxima and hence fail on extremely difficult problems, where a significant majority of workers get the answer wrong. This has caused some researchers to conclude that crowdsourcing can never achieve near-perfect accuracy, *no matter how many workers are involved* (Demartini, Difallah, and Cudré-Mauroux 2012).

We argue that part of the quality-control problem is an insistence on worker independence. Instead of isolating workers and blindly combining their votes, why not engage them to debate the question, seeking a consensus solution? Since Wikipedia's talk pages and dispute-resolution mechanisms have led to extremely high-quality content (Giles 2005), we adapt these mechanisms to the context of microtask crowdsourcing, creating the MicroTalk workflow.

MicroTalk starts by training workers. For the use case of producing training data for use by a machine learning algorithm, this involves presenting annotation guidelines and testing workers' responses on "gold standard" questions with known answers. Then, to get actual work done, MicroTalk composes three primitive microtasks — *Assess*, *Justify*, and *Reconsider* — to support asynchronous argumentation (Figure 1). In an Assess task, workers are asked to provide an answer to a given question. In a Justify task, workers provide reasoning with their answer in terms of the task guidelines taught during training. In a Reconsider task, workers are shown an argument for the *opposing* answer and then asked to reconfirm their original decision or change their answer. MicroTalk adaptively combines these tasks to collect a set of good justifications and arrive at a well-considered consensus answer. This paper makes the following contributions:

- We introduce micro-argumentation as a way to increase the accuracy of crowdsourcing on objective, consensus questions by having some workers justify their reasoning rather than just vote for an answer. When workers disagree, they are confronted with opposing arguments and given a chance to reconsider their assessment.
- We show that, by itself, argumentation is moderately effective at improving individual worker accuracy; but if one first filters the crowd to select workers producing better arguments ("discerning workers"), the approach works even better.
- We describe an adaptive, asynchronous workflow, MicroTalk, that combines Assess, Justify, and Reconsider tasks in a cost-effective manner. MicroTalk uses sim-

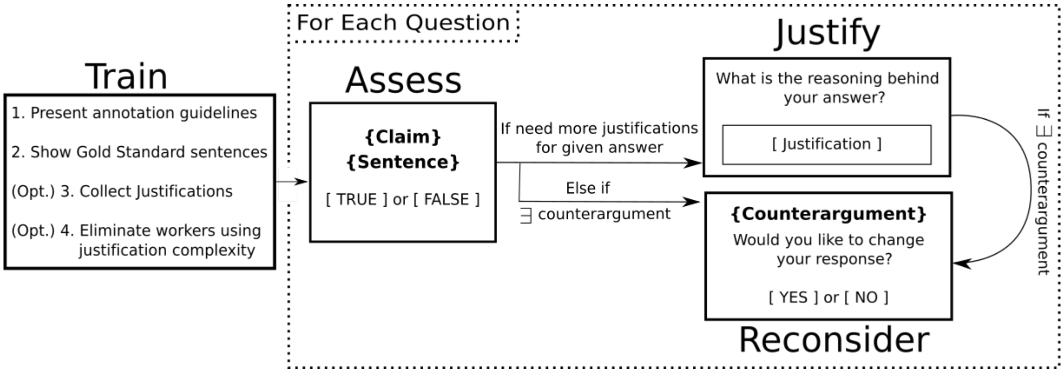

For Each Question
Train
1. Present annotation guidelines
2. Show Gold Standard sentences
(Opt.) 3. Collect Justifications
(Opt.) 4. Eliminate workers using justification complexity
Assess
{Claim}
{Sentence}
[TRUE] or [FALSE]
If need more justifications for given answer
Else if ∃ counterargument
Justify
What is the reasoning behind your answer?
[Justification]
If ∃ counterargument
{Counterargument}
Would you like to change your response?
[YES] or [NO]
Reconsider

Figure 1: For each question, MicroTalk issues workers an *Assess* task, presenting them a (claim, sentence) pair and prompting them to enter their answer. Some workers are then asked to provide an argument supporting their answer during a *Justify* phase. If there is a counterargument available, then a *Reconsider* task displays it, asking the worker if they wish to change their answer.

ple consensus on easy problems, requests justifications when disagreement occurs, identifies "discerning workers" who are likely to intelligently respond to arguments, and prompts them to consider alternative answers.

- We present experiments on an NLP relation-extraction task with workers from Mechanical Turk, showing (1) presenting arguments from automatically-identified "discerning workers" increases individual worker accuracy from 58% to 78%, and (2) controlling for cost, MicroTalk achieves 84% accuracy overall, compared to 64% for soft EM, and 58% for simple majority vote.

Previous Work

Virtually all previous approaches for addressing variability in worker skill ask multiple workers to perform the same (or related) tasks and then aggregate responses to infer the correct answers. Since the effectiveness of the method is highly dependent on the method for aggregating responses, numerous strategies have been investigated.

Most approaches assume that the question posed to crowdsourced workers is multiple-choice and has an objective answer, but workers may not answer correctly. The approaches further assume that the majority of workers are more likely to be correct than to make a mistake. Under these assumptions, Snow *et al.* (Snow et al. 2008) showed that one could often achieve high accuracy by exploiting redundancy via majority voting; indeed, this simple method can enable a crowd of ten novices to reach accuracies of 75% on NLP tasks such as sentiment analysis.

More robust methods learn worker skills. Rather than a simple majority vote, these approaches weight responses by using models of workers' abilities. The most common skill assessment method is very simple: into a mix of questions given to workers, include a random sample of *gold questions*, so named because the requester already has "gold standard" answers for these questions. Workers who fail to correctly answer these questions are dismissed or have their weights lowered. To avoid gaming behavior it is common to intermix questions with known and unknown answers. However, even this strategy is foiled by scammers building bots that utilize databases of known questions, necessitating elaborate strategies for programmatically generating an unbounded number of gold questions (Oleson et al. 2011).

More sophisticated approaches eschew gold questions, instead using unsupervised learning to jointly estimate worker accuracies and consensus answers. This method stems from Dawid and Skene (Dawid and Skene 1979), who consider a single question with an unknown correct answer and also parameters, $P_w(r|a)$, for each worker and each possible response, describing the probability that worker w will respond r when the true answer is a (a simple model of worker abilities). Subject to an important assumption, that workers' responses are conditionally independent of each other given the true answer, Dawid and Skene use *expectation-maximization* (EM) to estimate the latent question answers and worker accuracies. Whitehill *et al.* (Whitehill et al. 2009) note that worker responses are not really independent unless conditioned on both the correct answer *and* the question difficulty (another latent variable to be estimated). Welinder *et al.* (Welinder et al. 2010) take Whitehill's approach a step further, designing a model with general multi-dimensional parameters. Questions have many features, one of which could be difficulty, and workers are modeled as linear classifiers who make their responses by weighting those features. Kamar *et al.* (Kamar, Hacker, and Horvitz 2012) extract features from the task at hand and use Bayesian Structure Learning to learn the worker response model.

Unfortunately, all EM-based methods share several significant limitations. First, their greedy optimization is prone to local maxima; convergence is guaranteed, but not optimality. Specifically, if the data set includes difficult problems where a significant majority of workers gets the answer wrong, EM will likely converge to the incorrect answer. Indeed, these problems appear to surface often in practice. For example, Demartini *et al.* describe ZenCrowd, an elaborate method for named entity linking that combines crowdsourcing with probabilistic reasoning yet can't achieve precision higher than 85% no matter the number of crowd workers assigned (Demartini, Difallah, and Cudré-Mauroux 2012). In fact, the authors note “augmenting the number of workers performing a given task is not always beneficial.”

The methods described above are *post-hoc* — they are applied to a set of worker responses after they are gathered. A complementary set of approaches actively chooses which questions to ask which workers and how many times to ask each question. Karger *et al.* (Karger, Oh, and Shah 2011) algorithmically assign tasks to workers using a low-rank matrix approximation, but their method assumes that all tasks are equally difficult and workers are either perfectly correct or randomly guess. Dai *et al.* pose the worker control problem as a partially-observable Markov decision process (POMDP) showing substantial accuracy improvements for a given budget (Dai et al. 2013). Kamar *et al.* apply similar techniques to citizen-science applications (Kamar, Hacker, and Horvitz 2012). These methods are powerful, but they still use techniques like EM to combine worker assessments and hence are susceptible to local maxima. Furthermore, these methods keep workers isolated in order to maximize the independence of errors. As a result there is no way for an expert worker, who has correctly solved a problem, to convince other workers that her answer is correct.

Other forms of worker coordination can also be used to improve answer quality. Some complex problems can be solved by dividing work between several workers, through either sequential or simultaneous work (André, Kraut, and Kittur 2014). On certain problems, global constraints may be used to coordinate the activities of multiple workers (Zhang et al. 2012b). Unfortunately, some problems, like those we consider, are difficult to partition and may require careful reasoning in order to arrive at a correct answer. Kriplean *et al.* (Kriplean et al. 2014) built a system for public deliberation on election ballot measures, which uses pro/con points to help individual voters make informed decisions; by contrast, we seek to use deliberation to arrive a single objective correct answer. Question-and-answer sites like StackOverflow also produce answers to difficult problems, but can be tricked when the majority is wrong. A complementary form of worker coordination involves interactions between workers with various degrees of expertise, either directly (*e.g.*, through mentoring (Suzuki et al. 2016)) or indirectly (*e.g.*, through worked examples (Doroudi et al. 2016)), in order to improve the expertise of the worker pool. One could also seek to improve expertise by developing an effective procedure for training and testing workers (Liu et al. 2016).

Other researchers have developed methods for dealing with cases where the majority may be wrong, including tournament voting (Sun et al. 2011) and Bayesian truth serum (Prelec and Seung 2006), but it is not clear these methods work well for binary questions or that they will work on problems that require careful reasoning. Another line of work has shown that having annotators provide “rationales” for their answers by highlighting portions of text (Zaidan, Eisner, and Piatko 2007) or an image (Donahue and Grauman 2011) can improve machine learning classifiers. In contrast, we consider more complex arguments that can include information not found in the example to be classified, and show arguments to other workers rather than a machine. Finally, we note that incentives may also help to improve answers or arguments; Lasecki *et al.* perform quality control in a crowd-powered conversational agent by paying more to workers who propose or vote on winning answers (Lasecki et al. 2013).

LivedIn means that a person spent time in a place for more than a visit. Working in a location does *not* imply that a person has a LivedIn relation. However, you may assume that someone who has held a national office or played for a national sports team has lived in the country they serve or represent. Ambassadors should be counted as national-level officials. For example,

> Claim: *Ian Khama* ”lived in” *Botswana*
>
> Sentence: *Botswana*’s President *Ian Khama* is one of the few African leaders to openly criticize Mugabe.

Ian Khama holds national office for Botswana, therefore it can be concluded that he also lives in Botswana. The answer to this question is True.

Other Rules: You should only select relations that can be inferred by reading the sentence, even if you know others are true. You also should not select facts that are likely to be true, but you are not sure after reading the sentence.

Figure 2: Annotation guidelines for the LivedIn relation.

Relation Extraction Domain

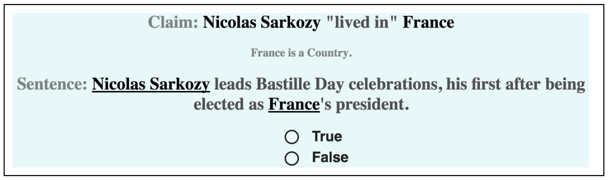
While our argumentation method is general, our evaluation of the method must be in a specific domain. We introduce that domain now, since it is helpful to use concrete examples as illustrations in the next section.

As our test domain we chose the high-level task of annotating training data for *relation extraction*, the task of generating structured information (relational tuples, such as would be found in a SQL database) from natural language text. Many researchers and practitioners are interested in relation extraction, and several recent efforts use machine learning approaches based on crowdsourced data acquisition (Zhang et al. 2012a; Angeli et al. 2014; Liu et al. 2016).

Specifically, we consider the problem of annotating a sentence to indicate whether it encodes the TAC KBP relation *LivedIn*. While determining whether a sentence supports the conclusion that a person lived (or lives) in a location might seem simple, the Linguistic Data Consortium annotation guidelines are surprisingly complex (Surdeanu 2013) and some sentences are tricky. For example, just because someone was born in a city or works there does not imply that they lived there. Figure 2 shows the instructions given to our workers. Figure 3 shows a *True* example.

Crowdsourcing Argumentation

In order to encourage discussion and reflection about the job being performed, MicroTalk combines several microtasks in an adaptive manner. We start by discussing high-level design decisions about the style of argumentation. We next explain MicroTalk’s three microtasks. We argue that some workers are better at argumentation than others and present a method for selecting these “discerning workers.” The section ends with a description of the complete MicroTalk workflow.



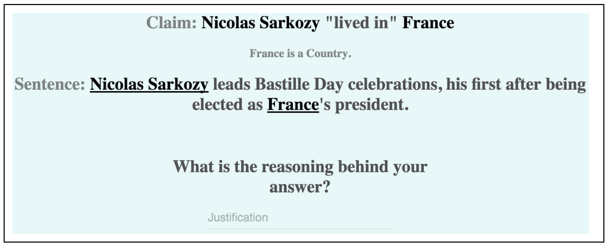
Claim: Nicolas Sarkozy "lived in" France

France is a Country.

Sentence: Nicolas Sarkozy leads Bastille Day celebrations, his first after being elected as France's president.

○ True
○ False

Figure 3: The Assess microtask.



Claim: Nicolas Sarkozy "lived in" France

France is a Country.

Sentence: Nicolas Sarkozy leads Bastille Day celebrations, his first after being elected as France's president.

What is the reasoning behind your answer?

Justification

Figure 4: The Justify microtask.

Synchronous *vs.* Asynchronous Argumentation

The most critical decision when trying to support automated argumentation is whether to implement a synchronous process, with multiple workers interacting in real time, or an asynchronous method, in which workers can only see the output of previous workers with no back and forth communication. Clearly, the synchronous context is closer to the forms of debate we find natural and the ability for participants to discuss each others points interactively would offer many advantages. Unfortunately, implementing such a real-time approach is expensive, requiring a retainer model or other incentives for rapid response (Bernstein et al. 2011; Lasecki and Bigham 2013). As a result, we defer synchronous argumentation to future work.

In this paper we focus on an asynchronous approach, which simplifies the worker recruitment model and also the design of the primitive microtasks, but poses its own challenges. Specifically, how can one best simulate a discussion when one participant leaves before the other arrives? Clearly, arguments must be cached, but the order in which they are collected may have a strong effect on workflow efficiency. We return to these questions after discussing the base units of our workflow.

Primitive Tasks

In order to collect high-quality data from crowd workers, MicroTalk combines worker training and qualification with three different microtasks: Assess, Justify, and Reconsider. Training is fairly conventional: MicroTalk displays a set of instructions and rules to the worker in order to explain the task of relation extraction. We determine if a worker is qualified to complete our tasks by asking them a set of gold questions with known answers. Workers who answer incorrectly are removed from the rest of the experiment but are paid for the short time they spent. Workers begin to see the other microtasks after completing the training step.

Each *Assess* task asks a worker a question. In our case, we ask them to assess whether a sentence states the LivedIn relation between two specific entities (e.g., see Figure 3).

A *Justify* task may be issued after a worker has completed an assessment. In it, the worker is simply asked to explain their reasoning using a text box with no enforced limit on length. Figure 4 shows an example. In preliminary studies on another NLP domain (named entity linking) we considered a variation of the Justify task that provided a more structured interface, requiring workers' arguments to reference numbered rules in the annotation guidelines. Our reasoning was that this approach might make it faster for workers to both construct and validate arguments, but the process proved cumbersome. When we redesigned the task, we aimed for maximum flexibility, inspired by lessons from Groupware (Grudin 1994), but as the underlying job (and hence, justifications) becomes increasingly complex, structured arguments may be worth revisiting in future work.

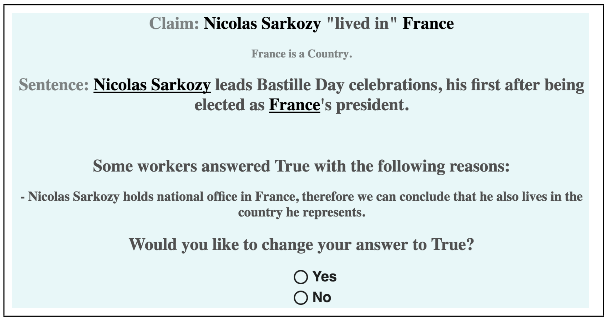
MicroTalk's final microtask is *Reconsider*, which may be issued when a worker selects an answer for which an opposing argument has been acquired in a previous justify task. In such a case, MicroTalk notifies the worker that someone disagrees with his or her decision and presents the counterargument. The worker is then given a chance to alter their previous decision for full "credit" on the task (Figure 5). Of course, there is no guarantee that the counterargument is correct, so the success of MicroTalk is (in part) dependent on its ability to identify either good justifications or discerning workers who will recognize their mistakes, but not be fooled by poor arguments. (More on this in the next section).

In early prototypes, we also considered additional task designs. For example, we thought workers might find it useful to be presented with dueling "pro" and "con" arguments during their *initial* assessment of a question; this led to the construction of an *Assess-w-arguments* microtask. Unfortunately, initial experiments worked poorly — workers were no more likely to select the correct answer when presented with arguments for both sides. We also considered microtasks that implemented an iterative improvement workflow (Little et al. 2009) for enhancing the quality of both "pro" and "con" justifications. Better justifications may cause workers to think more deeply about switching their answer; we further address this in the experiments section.

MicroTalk uses the Assess, Justify, and Reconsider tasks in different combinations depending on the current worker's assessment and prior workers' responses. Figure 1 summarizes the constraints governing the order in which the tasks may be invoked on any given question, and we discuss the ordering in more detail below.

Selecting Discerning Workers

When experimenting with an early prototype of MicroTalk, we were surprised that almost as many workers were convinced to switch *away* from a correct answer by a bad argument as were moved to fix their answer by sound reasoning. Since crowd workers have been shown to have highly heterogeneous skills at many domain-specific tasks, we wondered if some workers might be better at explicitly reasoning about the logic underlying their answers than others. We



Claim: Nicolas Sarkozy "lived in" France

France is a Country.

Sentence: Nicolas Sarkozy leads Bastille Day celebrations, his first after being elected as France's president.

Some workers answered True with the following reasons:

- Nicolas Sarkozy holds national office in France, therefore we can conclude that he also lives in the country he represents.

Would you like to change your answer to True?

○ Yes
○ No

Figure 5: The Reconsider microtask.

deemed these hypothetical savants "discerning workers" and explored methods for identifying them.

We conjectured that Law School Admission Test (LSAT) questions from the "Critical Reading" section might provide a good method for filtering workers, but initial experiments gave inconsistent and unpromising results.

Another method we tried was selecting workers who provide lengthy justifications for their decisions. Of course, there is no guarantee that verbose writers are actually better workers than the terse, and a percentage of lengthy justifications are indeed muddled. However, word count *has* been shown to be a surprisingly accurate measure of Wikipedia article's quality (Blumenstock 2008). Although this method worked well for selecting "discerning workers" in our experiments, it may not generalize to other task domains.

Instead, we settled on using Flesch-Kincaid readability tests to determine the complexity of a worker's argument (Kincaid et al. 1975). By asking workers to provide justifications on gold questions during the training phase, we can compute the median complexity of submitted justifications and retain only workers who provide explanations of at least a certain grade level. As we show in our Experiments section, specifically Figure 6, this approach is surprisingly effective.

Proactive *vs.* Lazy Justification

Given our Assess, Justify, and Reconsider microtasks, we now consider the best way to compose them. As we show in the Experiments section, Reconsider tasks produce higher worker accuracy than Assess tasks, but they are also more expensive.[1] If most questions are easy with no disagreement, then there will be no need to ever issue Reconsider tasks and hence no need to ask any worker to justify their decision. Thus, if most questions are easy, the workflow should be *lazy* and delay issuing Justify tasks until the first worker disagrees in their initial assessment. On the other hand, if one expects disagreement on most questions, then the workflow should *proactively* ask workers to justify their decisions so it is prepared to issue Reconsider tasks as soon as possible.

To visualize this tradeoff, consider two possible sequences, S and S', of five worker assessments, reflecting easy and hard questions respectively:

$$S = \langle T, T, T, T, T \rangle$$

$$S' = \langle T, T, F, F, T \rangle$$

Assume that the system only wishes to collect one justification for each disputed answer. If the workflow issues Justify tasks proactively, then S would generate the following sequence of tasks: $\langle A^1, J^1, A^2, A^3, A^4, A^5 \rangle$ where A indicates an Assess task, J signifies a Justify task, and the superscript indicates the worker performing the work. Proactive assignment of Justify tasks results in one unnecessary task for sequence S—the lack of disagreement renders the justification unnecessary, since no Reconsider is ever issued.

In contrast, proactively initiating Justify on S' would generate the following sequence of 10 microtasks: $\langle A^1, J^1, A^2, A^3, J^3, R^3, A^4, R^4, A^5, R^5 \rangle$ with three reconsiders. If the workflow had instead waited for disagreement before requesting justification, it would have produced this sequence: $\langle A^1, A^2, A^3, J^3, A^4, A^5, J^5, R^5 \rangle$. Note that this workflow is only able to issue one Reconsider task (*vs.* three) because it doesn't have the necessary justifications. If it continued to get workers whose initial assessments followed S' then it would continue with $\langle A^6, R^6, A^7, R^7 \rangle$, which requires 12 microtasks before getting three Reconsiders.

Given the prior probability that independent workers will disagree, one can calculate the expected cost of proactive *vs.* lazy justification. But qualitatively, it is clear that the lazy approach only makes sense when most problems are easy and workers agree. Furthermore, in the worst case proactive justification will only require one additional microtask per question, so that is the approach we adopt in MicroTalk.

The (Final) MicroTalk Workflow

Algorithm 1 summarizes the complete MicroTalk workflow operating on a single Boolean question. In order to qualify "discerning workers" a preprocessing phase (not shown) trains a sample of k workers and estimates their median justification complexity on gold standard problems. This median complexity is then used in the qualification step of Algorithm 1. As input the algorithm is given a budget; each call to Assess, Justify, and Reconsider decreases the remaining budget by C_a, C_j, and C_r, respectively.

Operation is straightforward. MicroTalk asks the first worker to answer the question and also to justify it, since no justification exists for her answer ($J_{a_1} =$ ' '). In subsequent iterations MicroTalk tests to see if a counterargument exists for each answer; $\overline{a_i}$ denotes the opposite answer from a_i so if $J_{\overline{a_i}} \neq$ ' ', then a counterargument exists.

The first worker who disagrees with previous answers not only provides a justification, but also is given the option to reconsider their answer. If this worker decides to switch their assessment, then their original argument is *not* kept as a justification for their original answer, because the worker was not confident enough in his or her own reasoning to value it higher than the presented counterargument.

After spending the budget on a mixture of Assess, Justify, and Reconsider microtasks, MicroTalk aggregates the

[1] In order to Reconsider, the worker must first perform Assess and another worker must have completed Assess and Justify.

```
Input  : A question $q$, budget $B$, and task costs
         $C_a, C_j, C_r$
Output: An answer $A(q)$
$j_T$ := $j_F$ := ‘ ’ ;
$b := 0$ ;
for $i := 1$, increase by 1, while $b < B$ do
    Justifying = F ;
    Train and qualify worker $w_i$;
    $a_i$ := Assess$(q, w_i)$ ;
    $b$ := $b + C_a$ ;
    if $j_{a_i}$ $=$ ‘ ’ then
        $j_{a_1}$ := Justify$(q, a_i, w_i)$ ;
        $b$ := $b + C_j$ ;
        Justifying := T
    end
    if $j_{\overline{a_i}} \neq$ ‘ ’ then
        $a$ := Reconsider$(q, j_{\overline{a_i}}, w_i)$ ;
        $b$ := $b + C_r$ ;
        if $a \neq a_i \wedge$ Justifying then
            $j_{a_i}$ := ‘ ’
        end
        $a_i$ := $a$
    end
end
return Aggregate$(a_1, \ldots, a_i)$;
```

Algorithm 1: The MicroTalk argumentation workflow.

workers' disparate opinions and returns the result. Majority vote is one way to aggregate assessments, but expectation maximization (EM) works better (as we show in the next section). Explicitly incorporating EM into our pseudocode would complicate the logic, since it requires reasoning across multiple questions.

Experiments

The experiments in this section address the following questions: 1) Are workers able to formulate a convincing argument with their assessment? 2) Does argumentation have an effect on individual workers' accuracy? 3) Do workers perform better when higher quality justifications are shown? 4) How do we find high-quality workers and how much better at argumentation are they? 5) How does the MicroTalk workflow compare with other approaches and is it cost effective?

Experimental Setup

To find a set of challenging TAC KBP questions, we considered worker annotations for a set of five TAC KBP person-place relations collected by previous researchers (Liu et al. 2016). We determined the most difficult sentences by ranking them according to the average L_1 distance between the boolean vector representing a worker's labeling of the five relations for that sentence and the boolean vector for the ground-truth labeling. Since 20 of the 25 highest ranked (most difficult) sentences were positive instances of the *LivedIn* relation, we used those for our experiments.

We ran experiments on Amazon Mechanical Turk, using workers who had completed at least 1,000 tasks with a 97% acceptance rate. As described earlier, the TAC KBP annotation guidelines for the LivedIn relation specify a number of counter-intuitive rules, which we make available to workers throughout each experiment as shown in Figure 2.

In each experiment, we first trained workers on the guidelines for annotating the LivedIn relation. Workers were given five gold standard questions and had to get three (60%) or more correct to complete the rest of the experiment. For this qualifier, we chose questions that had previously been shown to be the strongest indicators of whether a worker was likely to have a high or low overall accuracy. In the training phase, workers were asked to Assess and then Justify each gold standard question, in order to provide an indication of the quality of that worker's answers and the arguments they would be likely to supply.

Our workflows set the payment for each microtask (Assess, Justify, Reconsider) at $0.05 each. These rates were chosen such that the hourly payout was roughly equivalent across workers in each experiment. Workers were incentivized with bonuses for the Justify and Reconsider tasks — they were told they would receive a bonus of $0.05 for every high-quality argument they provided as well as a bonus of $0.05 if they chose the correct response for Reconsider.

Can Workers Generate Quality Justifications?

Since the success of the MicroTalk workflow depends on the ability of workers to write convincing arguments, our first experiment analyzed a sample of the arguments generated by workers to determine their quality. We found that workers provide arguments of highly varying quality, and that workers with higher overall accuracy tended to produce more convincing arguments.

As an example, consider the following sentence (with bolded entity mentions) taken from our pool of experimental questions:

> “The **United States** needs to be ready to press compromise proposals, something **Bush** and his secretary of state, Condoleezza Rice, show little interest in doing.”

We looked at top performing workers' justifications and compared these by hand to those of the least accurate workers. Justifications from accurate workers tend towards explicit reasoning, making reference to the annotation guidelines. Two example justifications are: “Bush may have been the leader of a different country other than the US that was the other partner in the attempted negotiation.” and “It is safe to assume that they are talking about President Bush in mentioning ‘his secretary of state’, the past president of the US, therefore he would have to live in the US.” The second justification explains that Bush must be a national official because he has a secretary of state, which justifies using the rule that national officials lived in a country. However, the first claims that this sentence could be referring to another pair of national officials that are not necessarily from the United States.

On the other hand, low-performing workers tended to favor less complex sentences and used logic that depends on information not found in the sentence. Two example justifications are: "He was president" and "I lived through that disaster. I can honestly say that, yes, Bush lived in this country and he was somehow president." The brevity of the first justification and the reasoning based on personal life experience in the second answer do not make convincing arguments for the correct answer.

In the following sections, we evaluate techniques for selecting workers who can improve their answer accuracy by reconsidering questions in response to high-quality arguments, and who are infrequently tricked by incorrect arguments.

Effect of Argumentation on Individual Workers

Our second experiment considers whether argumentation can improve the answer quality of individual workers. Workers were assigned to one of two conditions. In the baseline condition ($N = 51$ workers), workers were only presented with the Assess microtask for each question. In the other condition ($N = 116$ workers), workers participated in a non-adaptive workflow, which consisted of an Assess task, a Justify task, and a Reconsider task. The arguments presented in the Reconsider microtasks were chosen from a previous run of the same experiment, selecting those from the most accurate workers (the most accurate worker for each question was determined by that worker's accuracy on the remaining questions). This experiment used a subset of 10 out of the 20 candidate sentences.[2] As in all our experiments, workers were asked to first answer the 5 gold questions and they were paid \$0.05 per task completed.

Workers given all three microtasks were significantly more accurate than those that only completed Assess tasks. Accuracy in the baseline condition (Assess only) was 59%. Accuracy in the experimental condition (Assess, Justify, and Reconsider tasks) was 71%. This 12% accuracy improvement is statistically significant ($p = 0.0003$, $t = 3.695$), indicating that Justifying and Reconsidering answers improves worker accuracy. We also confirm that workers' accuracies are normally distributed by using a Shapiro-Wilk test for normality ($p > 0.05$) (Shapiro and Wilk 1965).

Some of the increase in accuracy appears to come simply from the act of justifying one's answers. Even ignoring the revised decisions from Reconsider tasks, workers had a 7% increase in accuracy, from 59% in the baseline condition (Assess task only) to 66% in the experimental condition. This difference is significant under a two-sided t-test ($p = 0.022$, $t = 2.318$).

We note that further experiments are required to completely determine whether the improvement in accuracy is due to the Justify or Reconsider tasks, or the combination. Previous research has shown that both self assessment (Dow et al. 2012) and comparing solutions to those provided by other workers (Mamykina et al. 2016) can improve answer quality. Dropout rates were not significantly different between the two conditions, but we are unable to rule out a worker selection effect. Either way, the inclusion of Justify and Reconsider jobs substantially improves answer quality compared to solitary Assess microtasks.

Effect of High-Quality Justifications on Workers

We also wanted to explore whether more convincing arguments written by experts would have an effect on how often workers changed their answer and the accuracy of their answers. To test this, we chose 5 sentences and launched a third experiment where each worker would be assigned a sequence of Assess and Reconsider microtasks. The counter-argument seen by workers during the Reconsider phase was randomly chosen: either an "expert" argument written collectively by the authors or a worker-generated justification from an earlier experiment.

Workers that saw arguments written by other workers changed their answer 20% of the time. However, when workers were shown arguments written by the authors, this rate increases to 46%. Simply put, arguments written by experts are significantly more convincing than arguments from workers, $\chi^2(1, N = 335) = 26.29,\ p < 0.00001$. There was no statistically significant difference in the number of responses that were changed to the correct answer in either condition, $\chi^2(1, N = 107) = 0.009,\ p = 0.924$. Although expert justifications made no difference in the percentage of responses changing to the correct answer, they were able to convince a significantly higher number of workers to change their answer. Reconsidered responses from workers that were shown worker-generated justifications had a mean accuracy of 71%, which is similar to the accuracy observed in the previous experiment. Reconsidered responses from workers that were shown arguments written by experts improved this accuracy by 9% from 71% to 80%; however, the difference is only marginally significant, $\chi^2(1, N = 335) = 3.023,\ p = 0.08$, perhaps due to the small sample size. We deem the results promising enough that we wish to consider a future version of MicroTalk which includes iterative improvement steps to hone the best worker justifications.

Benefits of 'Discerning Workers'

When experimenting with an early prototype, we found that many workers would switch *away* from the correct answer when shown a bogus justification. To combat this effect we filtered our crowd for "discerning workers" using justification complexity (specifically, the Flesch-Kincaid readability test (Kincaid et al. 1975)) as our selection metric. To select promising workers, we first computed the median readability score for workers' justifications on the gold-standard filtering questions. We then computed the number of times that a specific worker has a justification with a higher reading grade level than the median complexity, averaged over the five gold questions. Our analysis, shown in Figure 6, shows the strong effects of this filtering technique applied to workers from our second experiment (non-adaptive Assess, Justify, and Reconsider microtasks). Filtering has a strong positive effect on the reconsidered accuracy of workers (the magnitude of the effect is stronger for more stringent filtering) but almost no effect on the accuracy when the workers' changes are *not* taken into account.

[2] Since workers would be providing arguments and reconsidering for every question, we wanted to keep the total time of the experiment comparable to the other experiments.

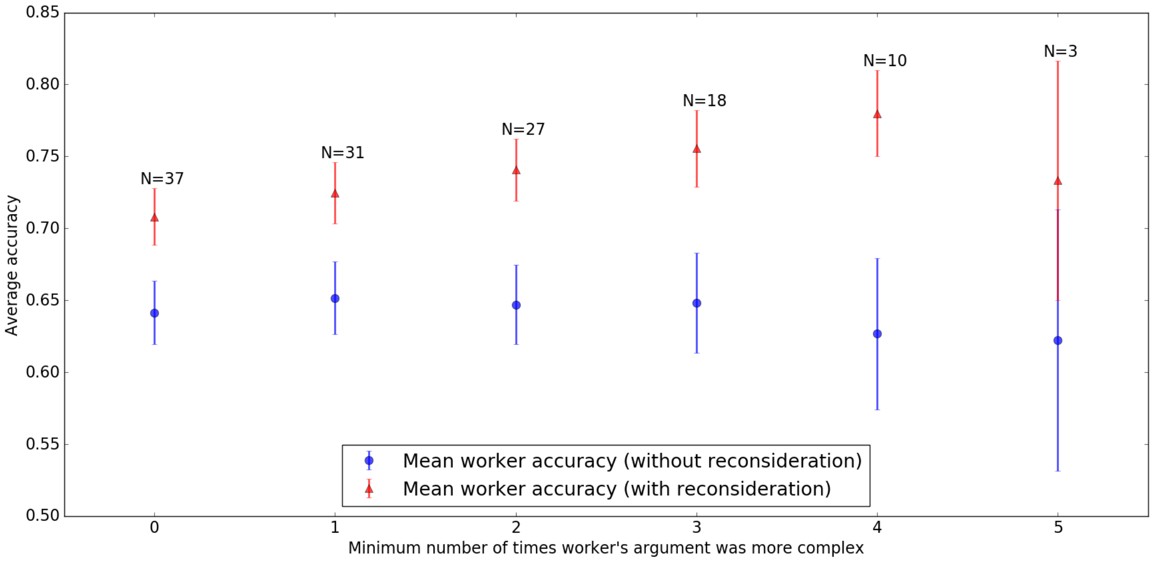


Figure 6: On average, workers were more accurate after reconsidering their answer in the context of opposing arguments (red points above blue). The X-axis shows the minimum number of times a worker provided an argument more complex than the median complexity on gold standard questions. Filtering workers with simple arguments does not affect the accuracy of their initial assessment (blue points trend flat as X increases), but it *strongly improves* the accuracy of reconsidered decisions (red points climb). $X = 0$ shows the crowd unfiltered, while $X = 5$ is only workers who had complex arguments on all gold standard questions. The number of workers, N, in each group is displayed as an annotation above the (red) reconsidered points.

We define discerning workers as workers with at least 4 ($\geq 80\%$) justifications with a higher Flesch-Kincaid score than the median reading grade level. The average accuracy of workers in this group is 63% without reconsideration and 78% with reconsideration. (Recall from the previous section that average accuracy with reconsideration for the general population is 71%, also shown at $X = 0$ in Figure 6.) In other words, filtering for discerning workers yields a 41% reduction in error. The significance of the improvement can be computed using a non-parametric test by bootstrapping a distribution of the overall mean accuracy for the discerning workers and the unfiltered crowd. Taking the paired difference (discerning workers — unfiltered crowd) between these accuracy distributions results in a distribution of the improvement in accuracy. Bootstrapping a distribution of 10,000 samples results in a mean improvement in accuracy of 6% with a significance test for difference: $\alpha = 0.026$.

It is also important to note that our experimental discerning worker filtering was done post-hoc, which means that discerning workers were responding to counterarguments that were written by non-discerning workers in most cases. Our result shows that even though discerning workers may be seeing lesser quality arguments, they are still able to make correct decisions about the arguments' correctness. An interesting direction for future work would be to see if the mean improvement in accuracy would increase further if discerning workers were constrained to only seeing justifications provided by other discerning workers.

Benefits of Complete Workflow

While we have shown that our argumentation workflow significantly improves individual worker accuracy, the additional cost of the workflow (*i.e.*, the Justify and Reconsider steps) may not be worth the improvement in answer quality. For instance, an alternative use of budget would be to show additional Assess tasks for each question and aggregate answers from a larger number of workers. In this section, we demonstrate that controlling for cost, the MicroTalk workflow out-performs this simpler approach.

We launched two different workflows for experimentation: *Simply Ask* and *MicroTalk*. In Simply Ask, workers did not provide or see justifications nor were they given Reconsider tasks. Workers were instead given a sequence of Assess microtasks. MicroTalk was run as previously described — all workers performed Assess and some were given Justify and Reconsider tasks according to the logic shown in Algorithm 1. Workers took a median time of 15 minutes to complete twenty sentences using Simply Ask, and a median time of 25 minutes using MicroTalk. The differences in time can be attributed to the extra tasks workers must complete in MicroTalk, which are not present in Simply Ask.

MicroTalk requires requesters to implement two additional microtasks over the Simply Ask workflow, which has workers only complete the Assess step. In total, our final experiment collected responses from 37 workers in our MicroTalk workflow and 68 workers for the Simply Ask baseline. In order to properly compare the two workflows, we fixed a budget for each question and simulated hiring workers using data from each. We gathered workers and varied

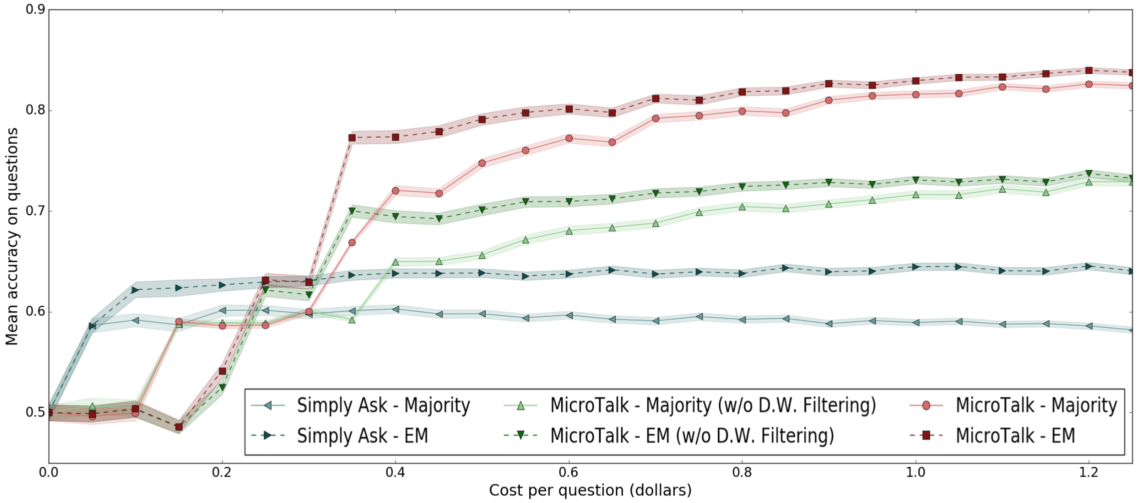


Figure 7: The best variant of our proposed MicroTalk workflow (MicroTalk - EM) significantly outperforms the best baseline (Simply Ask - EM), reducing mean question error by over 54%. The mean accuracies shown (with 95% confidence intervals) are computed over 1,000 runs of the workflow (randomizing worker order) on 20 questions.

the budget from \$0.00 until a maximum budget of \$1.20 was reached, since that is when the entire population of discerning workers was exhausted in the MicroTalk workflow. This was repeated 1,000 times per budget in different iterations, where the order of workers was randomized each time (as was the sample of workers since there is a fixed budget). The two workflows would run until the fixed budget was reached, at which point the iteration would terminate and question accuracy was calculated. For MicroTalk's iterations, only workers after the first disagreement had a randomized order because all workers before the disagreement had the same answer. The point of disagreement is different for every question in the experiment (and for both conditions) and is determined by the first worker to answer differently than all prior workers. All workers in Simply Ask had a randomized order since they were only completing an Assess task. We then plot the mean accuracy from each constrained budget that was tested to determine the accuracy of each workflow on our set of 20 questions.

We consider two versions of Simply Ask: one where decisions are aggregated with majority vote and one with expectation maximization (EM). With MicroTalk, we also show performance with and without discerning worker filtering. Figure 7 plots the results. Simply Ask performs similarly to previous baseline experiments with an average worker accuracy of 58%. Every workflow has an expected accuracy of 50% at $X = 0$, since a budget of zero means there are no workers in the worker pool, and aggregation reduces to random guess. As the budget of each problem is increased, the effects of argumentation become more apparent. MicroTalk (without discerning worker filtering) performs as well as Simply Ask at $X = 0.3$, which translates to 6 workers in Simply Ask. The effects are stronger in MicroTalk where workers are filtered to be discerning workers. At $X = 0.4$ (8 workers in Simply Ask), MicroTalk is able to achieve an average accuracy of 77% compared with Simply Ask (EM) at 64%.

We note that workers in Simply Ask converge to an answer almost immediately with only 3–4 workers. By increasing the budget and hiring more workers, no new information is gained, and the majority is still incorrect; thus, Simply Ask never achieves accuracy greater than 65% in our challenging domain. (A similar result was reported previously by (Demartini, Difallah, and Cudré-Mauroux 2012).) In contrast, MicroTalk produces an 84% accuracy, achieving an error reduction of over 54%.

Conclusion

Instead of isolating workers, we argue that they should debate their decisions. As a first step towards this vision, we introduce a novel, asynchronous workflow, MicroTalk, which prompts workers to justify their assessments and confronts them with counterarguments, allowing them to reconsider their decisions. While argumentation improves the accuracy of most people, it is especially successful for the subset of workers who write longer and grammatically sophisticated justifications for their own reasoning, a class we term "discerning workers." We experiment on the NLP task of relation extraction; our most significant results show (1) presenting arguments from automatically-identified "discerning workers" increases individual worker accuracy from 58% to 78%, and (2) controlling for cost, the MicroTalk workflow achieves 84% accuracy overall, compared to 58% from simple majority vote and 64% from soft EM.

Our future work includes applying MicroTalk to additional tasks, hopefully demonstrating its generality. To further improve the quality of results, we would like to try offering financial incentives for convincing arguments. We also hope to ask multiple workers to iteratively improve prior justifications, honing the best arguments. Alternatively, one could implement a micro-version of an explicit coordination model like that seen on Wikipedia (Kittur and Kraut 2008). We suspect that decision-theoretic methods, like those of (Dai et al. 2013), would improve efficiency. Perhaps most interesting would be to explore *synchronous* workflows for argumentation. Challenges would include mechanisms for maintaining a critical mass of real-time workers (Bernstein et al. 2011), and some way to ensure that discussion stays focused on the task at hand rather than diverging towards entertainment.

Acknowledgements

Shih-Wen Huang, Mausam, Chris Lin, and the anonymous reviewers gave extremely helpful feedback; Chris Lin and Angli Liu kindly provided relation extraction data. We thank the crowd workers who made this work possible and note NSF grant IIS-1420667, ONR grant N00014-15-1-2774, the WRF/Cable Professorship, and a gift from Google.

References

André, P.; Kraut, R. E.; and Kittur, A. 2014. Effects of Simultaneous and Sequential Work Structures on Distributed Collaborative Interdependent Tasks. In *CHI*, 139–148.

Angeli, G.; Tibshirani, J.; Wu, J.; and Manning, C. D. 2014. Combining distant and partial supervision for relation extraction. In *EMNLP 2014*, 1556–1567.

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *UIST*.

Blumenstock, J. E. 2008. Size matters: word count as a measure of quality on wikipedia. In *WWW 2008*, 1095–1096.

Dai, P.; Lin, C. H.; Mausam; and Weld, D. S. 2013. Pomdp-based control of workflows for crowdsourcing. *Artificial Intelligence* 202:52–85.

Dawid, A., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics* 28(1):20–28.

Dekel, O., and Shamir, O. 2009a. Good learners for evil tecahers. In *ICML*.

Dekel, O., and Shamir, O. 2009b. Vox populi: Collecting high-quality labels from a crowd. In *COLT*.

Demartini, G.; Difallah, D. E.; and Cudré-Mauroux, P. 2012. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW 2012*.

Donahue, J., and Grauman, K. 2011. Annotator rationales for visual recognition. In *ICCV 2011*.

Doroudi, S.; Kamar, E.; Brunskill, E.; and Horvitz, E. 2016. Toward a Learning Science for Complex Crowdsourcing Tasks. In *CHI*.

Dow, S.; Kulkarni, A.; Klemmer, S.; and Hartmann, B. 2012. Shepherding the crowd yields better work. In *CSCW 2012*.

Giles, J. 2005. Internet encyclopaedias go head to head. *Nature*.

Grudin, J. 1994. Groupware and social dynamics: Eight challenges for developers. *Commun. ACM* 37(1):92–105.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Conference on Communication, Control, and Computing*.

Kincaid, J. P.; Fishburne Jr, R. P.; Rogers, R. L.; and Chissom, B. S. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document.

Kittur, A., and Kraut, R. E. 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *CSCW 2008*.

Kriplean, T.; Bonnar, C.; Borning, A.; Kinney, B.; and Gill, B. 2014. Integrating on-demand fact-checking with public dialogue. In *CSCW*, 1188–1199.

Lasecki, W. S., and Bigham, J. P. 2013. Interactive crowds: Real-time crowdsourcing and crowd agents. In *Handbook of Human Computation*. 509–521.

Lasecki, W. S.; Wesley, R.; Nichols, J.; Kulkarni, A.; Allen, J. F.; and Bigham, J. P. 2013. Chorus: A Crowd-Powered Conversational Assistant. In *UIST*, 151–162.

Little, G.; Chilton, L. B.; Goldman, M.; and Miller, R. C. 2009. TurKit: Tools for Iterative Tasks on Mechanical Turk. In *HCOMP*.

Liu, A.; Soderland, S.; Bragg, J.; Lin, C. H.; Ling, X.; and Weld, D. S. 2016. Effective crowd annotation for relation extraction. In *Proceedings of NAACL and HLT 2016*.

Mamykina, L.; Dimond, J.; Smyth, T.; and Gajos, K. Z. 2016. Learning From the Crowd : Observational Learning in Crowdsourcing Communities. In *CHI*.

Oleson, D.; Sorokin, A.; Laughlin, G. P.; Hester, V.; Le, J.; and Biewald, L. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *Human Computation Workshop*.

Prelec, D., and Seung, S. 2006. An algorithm that finds truth even if most people are wrong. Working paper.

Shapiro, S. S., and Wilk, M. B. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52(3/4):591–611.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 254–263.

Sun, Y.-A.; Dance, C. R.; Roy, S.; and Little, G. 2011. How to assure the quality of human computation tasks when majority voting fails. In *Workshop on Computational Social Science and the Wisdom of Crowds, NIPS*.

Surdeanu, M. 2013. Overview of the TAC2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC 2013*.

Suzuki, R.; Salehi, N.; Lam, M. S.; Marroquin, J. C.; and Bernstein, M. S. 2016. Atelier: Repurposing Expert Crowdsourcing Tasks as Micro-internships. In *CHI 2016*.

Wauthier, F. L., and Jordan, M. I. 2011. Bayesian bias mitigation for crowdsourcing. In *NIPS*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Whitehill, J.; Ruvolo, P.; Bergsma, J.; Wu, T.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*.

Zaidan, O. F.; Eisner, J.; and Piatko, C. D. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Proceedings of NAACL and HLT 2007*.

Zhang, C.; Niu, F.; Ré, C.; and Shavlik, J. 2012a. Big data versus the crowd: Looking for relationships in all the right places. In *ACL*.

Zhang, H.; Law, E.; Miller, R.; Gajos, K.; Parkes, D. C.; and Horvitz, E. 2012b. Human computation tasks with global constraints. In *CHI 2012*.