Cicero: Multi-Turn, Contextual Argumentation for Accurate Crowdsourcing

Quanze Chen
University of Washington
Seattle, Washington
cqz@cs.washington.edu

Jonathan Bragg[*]
University of Washington
Seattle, WA
jbragg@cs.washington.edu

Lydia B. Chilton
Columbia University
New York, NY
chilton@cs.columbia.edu

Daniel S. Weld
University of Washington
Seattle, WA
weld@cs.washington.edu

ABSTRACT

Traditional approaches for ensuring high quality crowdwork have failed to achieve high-accuracy on difficult problems. Aggregating redundant answers often fails on the hardest problems when the majority is confused. Argumentation has been shown to be effective in mitigating these drawbacks. However, existing argumentation systems only support limited interactions and show workers general justifications, not context-specific arguments targeted to their reasoning.

This paper presents Cicero, a new workflow that improves crowd accuracy on difficult tasks by engaging workers in multi-turn, contextual discussions through real-time, synchronous argumentation. Our experiments show that compared to previous argumentation systems which only improve the average individual worker accuracy by 6.8 percentage points on the Relation Extraction domain, our workflow achieves 16.7 percentage point improvement. Furthermore, previous argumentation approaches don't apply to tasks with many possible answers; in contrast, Cicero works well in these cases, raising accuracy from 66.7% to 98.8% on the Codenames domain.

[*]Now at Stanford University

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300761

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; **Collaborative and social computing systems and tools**; *Empirical studies in collaborative and social computing*; Collaborative interaction.

KEYWORDS

Crowdsourcing; argumentation; dialog

ACM Reference Format:
Quanze Chen, Jonathan Bragg, Lydia B. Chilton, and Daniel S. Weld. 2019. Cicero: Multi-Turn, Contextual Argumentation for Accurate Crowdsourcing. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3290605.3300761

1 INTRODUCTION

Crowdsourcing has been used for a wide variety of tasks, from image labeling to language transcription and translation. Many complex jobs can be decomposed into small microtasks [2, 6, 26, 31]. After such decomposition, the primary challenge becomes ensuring that independent individual judgments result in accurate global answers. Approaches ranging from aggregation via majority vote [38] to programmatic filtering via gold-standard questions [32] have all been created to achieve this goal. Further improvements have led to more intelligent aggregation such as expectation maximization (EM) [8, 41, 42]. However, EM may still fall short, especially on hard problems where individual judgments are unreliable. Indeed, some researchers have concluded that crowdsourcing is incapable of achieving perfect accuracy [9].

Yet recently, *argumentation* has been shown to be an effective way to improve the accuracy of both individual and aggregate judgments. For example, Drapeau *et al.*'s MicroTalk [12] used a pipelined approach of: 1) asking crowd workers to *assess* a question's answer, 2) prompting them to *justify* their reasoning, 3) showing them counterarguments

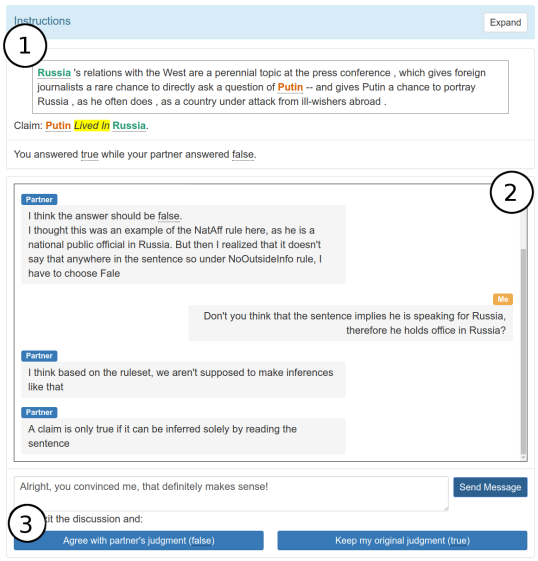

Figure 1: Discussion interface for use in Cicero, inspired by instant-messaging clients, showing a fragment of an actual discussion in the Relation Extraction domain. (1) Presents the question (sentence + claim) and both sides' beliefs. (2) Initial discussion is seeded with the workers' justifications. (3) Options added to facilitate termination of a discussion once it has reached the end of its usefulness.

written by other workers, and 4) allowing them to *reconsider* their original answers to improve individual judgments. In principle, this simplified form of argumentation allows a single dissident worker, through force of reason, to steer others to the right answer. Furthermore, the authors showed that argumentation was compatible with EM; combining the two methods resulted in substantial gains in accuracy.

However, while asynchronous argumentation systems like MicroTalk attempt to resolve disagreement, the steering power of a one-round debate is limited. Workers are only shown a pre-collected justification for an opposing answer; they aren't challenged by a specific and personalized argument against the flaws in their original reasoning. There is also no back-and-forth interaction that could illuminate subtle aspects of a problem or resolve a worker's misconceptions — something which may only become apparent after several turns of discussion. Furthermore, since justifications are pre-collected, workers need to write a generic counter argument; while this works for binary answer tasks, it is completely impractical for tasks with many answers; such a counter-argument would typically be prohibitively long, refuting $n-1$ alternatives.

This paper presents Cicero, a new workflow that engages workers in *multi-turn and contextual* argumentation to improve crowd accuracy on difficult tasks. Cicero selects workers with opposing answers to questions and pairs them into a discussion session using a chat-style interface, in which they can respond to each other's reasoning and debate the best answer (Figure 1). During these exchanges, workers are able to write context-dependent counter-arguments addressing their partner's specific claims, cite rules from the training materials to support their answers, point out oversights of other workers, and resolve misconceptions about the rules and task which can impact their future performance on the task. As a result of these effects, workers are more likely to converge to correct answers, improving individual accuracy. Our experiments on two difficult text based task domains, relation extraction and a word association task, show that contextual multi-turn discussion yields vastly improved worker accuracy compared to traditional argumentation workflows.

In summary, we make the following contributions:

- We propose Cicero, a novel workflow that induces multi-turn and contextual argumentation, facilitating focused discussions about the answers to objective questions.
- We introduce a new type of worker training to ensure that workers understand the process of argumentation (in addition to the task itself) and produce high quality arguments.
- We develop Cicero-Sync, a synchronous implementation of our workflow using real-time crowdsourcing, and apply it to conduct the following experiments:
  - In the Relation Extraction domain introduced by MicroTalk [12], we show that contextual, multi-turn argumentation results in significantly higher improvement in accuracy: a 16.7 percentage point improvement over individual workers' pre-argumentation accuracy *v.s.* a 6.8 point improvement using MicroTalk's one-shot argumentation. When aggregating the opinions of multiple workers using majority vote or EM, we see 5 percentage points higher aggregate accuracy, accounting for cost.
  - Using a version of the Codenames domain [47], that has many answer choices (making MicroTalk's non-contextual argumentation untenable), we show that Cicero is quite effective, improving individual worker accuracy from 66.7% to a near-perfect 98.8%.
  - We qualitatively analyze the discussion transcripts produced from our experiments with Cicero-Sync, identifying several characteristics present in contextual, multi-turn argumentation.

2 PREVIOUS WORK

Quality control has been a central concern in crowdsourcing, but space constraints preclude a complete discussion of the various post-hoc methods such as majority vote [38],

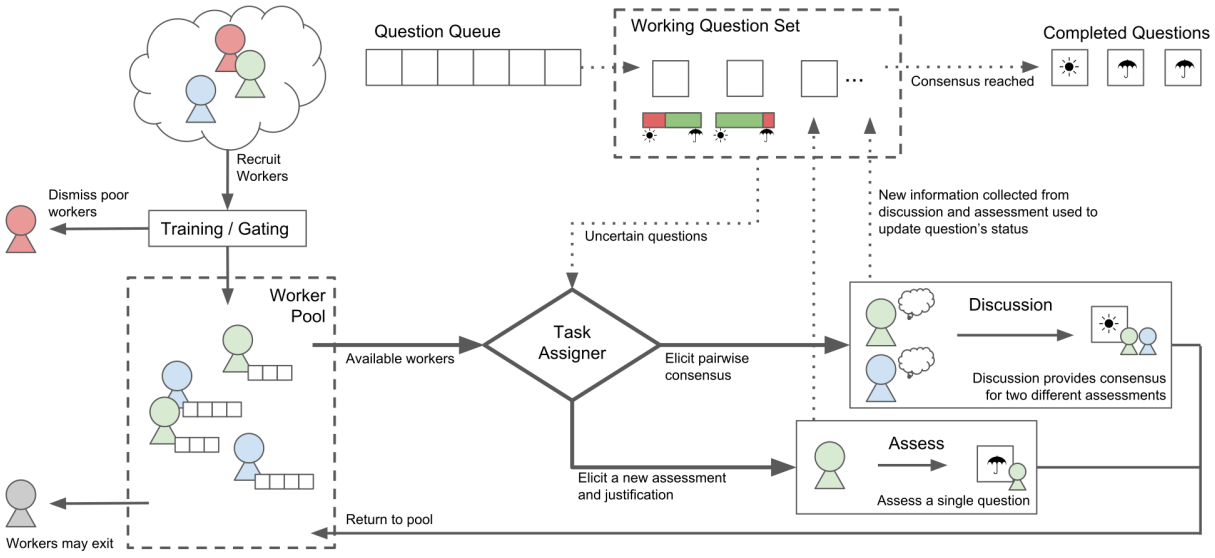

Figure 2: CICERO **System Diagram. Solid arrows indicate paths for workers through the system. Dotted arrows indicate how questions are allocated through the system.**

gated instruction [27], and programmatic generation of gold-standard questions [32]. Similarly, approaches to improve accuracy by assigning certain questions to specific workers [7, 18, 19] have also been suggested.

Expectation maximization [8, 41, 42] is especially popular, but all methods embody greedy optimization and hence are prone to local maxima. If the data set includes challenging problems, where a significant majority of workers get the answer wrong, EM likely converges to the incorrect answer.

Other researchers have investigated methods to handle cases where the majority may be wrong, *e.g.*, tournament voting [39] and Bayesian truth serum [34]. Unfortunately, these methods are unlikely to work for difficult questions where individual answers require detailed analysis.

Rationales & Feedback Can Improve Accuracy

Researchers have demonstrated that requiring annotators to submit "rationales" for their answers by highlighting portions of text [29, 44] or an image [10] can improve machine learning classifiers. However, rationales alone are insufficient for addressing any disagreement.

Dow *et al.* [11] conducted experiments demonstrating that timely, task-specific feedback helps crowd workers learn, persevere, and produce better results. Benefits of feedback also extend to that from peers: Ho *et al.* [14] show that peer communication improves work quality while Kobayashi *et al.* [20] demonstrate that reviewing can help workers self-correct. Additionally, Zhu *et al.* [48] noted that workers who review others' work perform better on subsequent tasks.

Crowd Deliberation and Debate

Crowd deliberation has been shown to be useful when disagreement is likely. Wiebe *et al.* [43] showed (with small-group, in-person studies) that there are benefits to getting annotators to reconsider their positions and discuss them with other workers. ConsiderIt [21] takes this kind of principled debate online and into the political spectrum, using pro/con points and encouraging participants to restate alternative positions to help voters make informed choices.

These papers inspired the design of MICROTALK [12], a workflow for one-shot argumentation, comprising of three microtasks — assess, justify & reconsider — asking crowd workers to assess a question, prompting them to justify their answer, and then encouraging them to reconsider (one-shot argumentation) their original decision given another worker's argument. Our work, CICERO, extends MICROTALK with multi-turn, contextual argumentation, which yields more accurate results.

Very recently, Schaekermann *et al.* [37] propose a multi-turn workflow for group deliberation and investigate factors that contribute to consensus. In contrast to Schaekermann *et al.*: 1) we introduce designs to train and test workers so they can argue effectively, 2) we support contextual communication to address scalability of justifications in multiple choice question domains, and 3) we use dynamic matching to expose workers to diverse counter-arguments and adapt to worker entry and drop-out.

3 CICERO DESIGN

In this section, we present the Cicero workflow as well as design considerations in a synchronous implementation of the workflow used for our experiments. We first explain the rationale for contextual, multi-turn discussions and give an overview of our Cicero workflow. We then talk about the decision to implement our workflow in a synchronous system—Cicero-Sync. Finally, we discuss the design choices we made to (1) create an interface for effective real-time discussion, as well as (2) improve instructions and training for the domains we examined.

Contextual and Multi-Turn Discussion

In natural forms of debate, participants who disagree take turns presenting arguments which can refute or supplement prior arguments. Our Cicero workflow is designed around the concept of emulating this process in a crowd work setting by using paired discussions facilitated by a dynamic matching system. Participants are matched with partners based on their current beliefs and are encouraged to present their arguments over multiple turns.

While real-life debates may include multiple participants each responsible for addressing arguments on different aspects of a problem, in the crowd setting we can utilize the diversity of workers to cover a broad set of views and reasoning; thus, to simplify the process, we focus on a two-participant discussion model.

Workflow Overview

Since argumentation happens on an ad-hoc basis, it's much more flexible to have our workflow focus on managing transitions between different states a worker may be in instead of defining a single pipeline. Due to this, our design of the Cicero workflow follows an event-based definition model where the automatic task assigner allocates tasks as workers' state changes. Figure 2 summarizes how our workflow allocates worker resources and questions in a dynamic way.

Initially, workers are recruited from a crowd work platform (such as Amazon Mechanical Turk) and are immediately assigned to a **training** task. Workers who pass training and the associated gating tests [27] enter the *worker pool* and wait to be assigned work. Then, instead of a fixed workflow, our event-based automatic task assigner decides which type of task and question to assign to a worker subject to a set of constraints. As workers complete their tasks and update the beliefs of questions in the working set, new candidate tasks are dynamically selected and allocated. Cicero's dynamic matching engages workers across diverse pairings, which has been shown to promote better output in large creative tasks such as in Salehi *et al.* [36].

In Cicero, there are two main types of tasks that the automatic assigner may assign to an idle worker: **assess** and **discussion**.

- The **assess** task acquires one worker from the worker pool who is then presented with one question — in our case a single question in the domain — that asks for an answer to a multiple choice question and optionally a free-form justification for their position. This task is a combination of the assess and justify microtasks in MicroTalk [12] as a single task.
- The **discussion** task acquires two workers from the worker pool who are both shown a discussion interface for a question. At the end of a discussion, the justification text may be updated for both workers. This task is a multi-turn, contextual version of the reconsider task in Microtalk [12], which actively engages both workers. We will cover details on the design of the discussion task in later sections.

The automatic task assigner is defined as a policy that decides which type of task should be allocated when a worker changes their state (such as upon completing a micro-task) and, depending on domain, can be designed to prioritize specific kinds of tasks, particular questions or qualities such as minimizing worker wait time and increasing concurrent work.

In general, the task assigned can be adapted to the goals of the requester. However, there are a few general constraints that the task assigner must follow:

- **Incompatible beliefs**: A discussion may only be assigned to workers if they have incompatible beliefs. Implicitly, this also requires existence of the both beliefs, implying they must have been collected (*e.g.*, via assess tasks).
- **No repeated discussions**: Two workers may only discuss a question if they have never discussed the question with each other before.

These constraints guarantee that the workflow will eventually terminate when there are no more workers who disagree and have never paired with each other.

There are many benefits to dynamically allocating partners. Since pairings aren't fixed, Cicero can automatically adapt to existing workers dropping out and new workers entering the pool. Additionally, in contrast to previous systems [12, 37], Cicero's automatic task assigner sequentially exposes each worker to discussions with multiple partners for a particular question. This allows for the possibility of a minority opinion reaching and convincing the majority. A worker who is convinced by a minority belief is able to spread the new answer as they may now be matched with workers they used to agree with, increasing the size of the minority.

CICERO-SYNC: A Real-Time Implementation

While the CICERO workflow does not constrain the type of interaction during a discussion task, we decided to test out the effectiveness of our workflow using synchronous discussions. A synchronous and real-time discussion environment allows us to mimic real world continuous dialogue spanning many turns thus preserving discussion context in a simple and natural way.

In CICERO-SYNC, workers are held in a waiting room until a partner becomes available. Once workers are matched into a discussion, they will not be assigned other tasks for the duration of that discussion and are expected to give each other their undivided attention. We note that, while useful for experiments, this design has limitations: the synchronous nature of discussions means that some workers will have to wait for a partner to become available and workers need to be online and active within the same time window, both of which imply a higher cost to the requester.

Additionally, there are many practical challenges to implementing and setting up synchronous real-time experiments with crowd workers, including implementing real-time client-server communication and working with APIs for worker recruitment and payment [16]. Fortunately, there have been enough real-time, crowd deployments [2, 3] that many useful lessons have been distilled [15]. We elected to use TurkServer [28], whose tools simplify the interfacing with Amazon Mechanical Turk for worker recruitment and task management and allow us to automatically track worker state as well as building our worker pool (Figure 2) using the TurkServer *lobby*.

Discussion Interface

The discussion task is the most important and defining task of the CICERO workflow. We considered multiple different options for the discussion interface focusing on ways to organize discussion structure and facilitate discoverability.

Early proposals included designs that were inspired by the posts-and-replies interfaces in social network timelines and the split-view pros-and-cons interfaces used in ConsiderIt, a political, argumentation system [21]. Our pilot studies showed that these methods were cumbersome and non-intuitive, so we decided on a free-form instant messaging (chat) metaphor for the discussion task (shown in Figure 1).

When a pair of workers enter a discussion, they are placed into a familiar instant messaging setting, where they can freely send and receive messages. Each message is tagged with being either from the worker themself ("me") or their unnamed partner ("partner"). An additional "exit" section below the chat interface allows either participant to terminate the discussion if they feel that it is no longer useful. Workers can utilize this exit mechanism to indicate that a consensus was reached or that no agreement is possible between them.

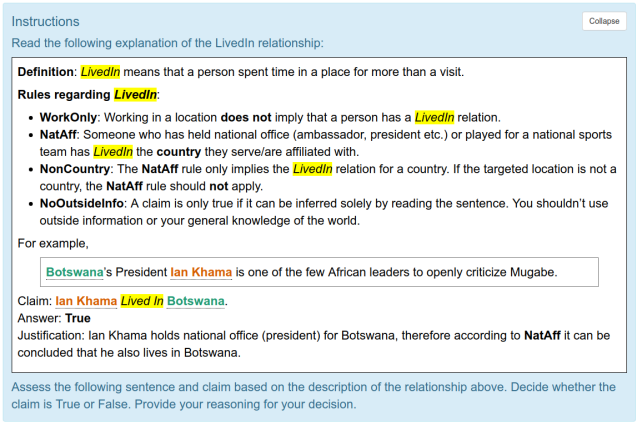

Figure 3: Screenshot of our *LivedIn assess* task (Relation Extraction domain) instructions containing 5 citable rules including the definition. Shorthands (in bold) allow for efficient citation of rules during discussion and within justifications (as shown in the example's justification).

The discussion interface can be easily adapted to specific needs of each experiment domain: In the Relation Extraction domain, the justifications collected from earlier assess or discussion tasks are used to seed the system, which we found to be beneficial in starting a conversation. In the Codenames domain, a drop-down menu below the text input field accommodates switching to alternate answers during the discussion addressing the non-binary nature of the questions.

Optimizing Task Instructions

Good instructions are essential for high inter-annotator agreement [27]. We observed in early pilot experiments that arguments which refer explicitly to parts of task guidelines were more effective at convincing a partner. However, the original task guidelines and training did nothing to encourage this practice. Workers came up with different ways to refer to parts of the instructions or training examples, but this was inconsistent and frequently caused confusion. References to the guidelines were hard to identify making it harder for workers to determine correct invocations of rules in the Relation Extraction domain pilots. Since arguing in synchronous discussion sessions is time-sensitive, creating rules and shorthands that are easy to cite is important for discussion efficiency.

We adjusted the task guidelines for the Relation Extraction domain from those in MICROTALK, re-organizing them into five concrete and easy-to-cite rules as shown in Figure 3. Each rule was given a shorthand so that workers can

unambiguously refer to a specific rule and aid in identification of proper or improper rule usage during the discussions. We observed that citing behavior became more consistent within discussions with workers frequently utilizing our shorthands in the discussion context. In the Codenames domain, which has simple instructions but a lot of example cases, we designed the instructions to both show the general guidelines and also provide a way for workers to review examples from training if they decide to reference them.

Selecting and Training Effective Workers

In initial pilots with Cicero-Sync, we noticed that workers were performing inconsistently. Following Drapeau *et al.*, we tried filtering for "discerning workers" using the Flesh-Kincaid score [17] to eliminate workers whose gold-question justifications were poorly written; to our surprise, this did not increase worker quality, but it did substantially reduce the number of possible workers. Filtering workers based only on gold standard question performance was also ineffective as it did not train workers to understand the rules required for our complex tasks.

Instead, we implemented a gating process [27], that can both train and select workers at the same time. Workers are presented with questions laid out in a quiz-like format. Each training question is provided along an introduction of related concepts from the task instructions. The questions are interleaved with the instructions in an interactive tutorial where new questions are presented as new concepts are introduced to reinforce worker understanding. Automated feedback is given when a worker selects an answer. At the end, workers' performance on a set of quiz questions is recorded. If a worker's accuracy on the quiz falls below a certain threshold, the worker will be asked to retry the training section (a limited number of times) with the order of the quiz questions randomized. Workers are dismissed if they exceed the retry limit.

Selecting and Training Effective Argue-ers

In existing argumentation systems [12, 37], training focuses on the target task instructions, however, not all kinds of arguments are productive. Argument forms and norms that contribute to positive discussion have been studied in the education community, termed 'accountable talk' [30]. During our pilot studies, we found that many workers' arguments weren't accountable, and realized that we need to train workers *how to argue* in order to ensure that discussions between workers are productive. To address this, we designed a novel **justification training** task incorporated as a part of the training process to train the workers to recognize good justifications and arguments before they interact with a partner.

In this training task, workers encounter a sample assess task, followed by a justification-like task where, instead of a free-form justification, workers are asked to select the best one from a list. We then provide feedback in the form of an argument for why a justification is better or worse with reference to the task rules. By undergoing this training, workers are exposed to both how to think about justifications and what an effective counter-argument can be.

In the Relation Extraction domain, specifically, each incorrect option targets a potential pitfall a worker may make when writing a justification, such as: failure to cite rules, incomplete or incorrect references to the rules, or making extended and inappropriate inferences. In the Codenames domain, questions can have ten or more possible answers, so it's not practical to create and present multiple justifications for all of them. Therefore, the training is adjusted to instead show a reference counter-argument when a worker selects an incorrect answer that refutes the incorrect choice and supports a correct one. Our sample questions are designed to illustrate different argumentation strategies in different situations as the rules in this domain are simpler.

We note that this design of exposing the concept of arguments to workers during training can be generalized to many domains by providing feedback in the form of counter-arguments. By training workers to recognize and analyze arguments (before they enter a live discussion), our justification training promotes more critical discussion.

Worker Retention and Real-Time Quality Control

Due to the synchronous nature of discussions in Cicero-Sync, workers may become idle for short periods of time when they are waiting to be matched to a partner. To ensure that idle workers in the *worker pool* are available for future matching, we implemented a real-time *lobby* design where workers wait while a task is assigned. This design was mainly inspired by both the default lobby provided in TurkServer [28] and from a worker-progress feedback design developed by Huang *et al.* [15] for low-latency crowdsourcing. While in the lobby, workers are presented with information on their peers' current status, such as how many workers are currently online and which workers may become available soon. Workers also see statistics on their work, which is tied to bonus payments, and are encouraged to wait. In Cicero-Sync, the task assigner is configured to immediately assign work as it becomes available. While in the lobby, a worker can voluntarily exit with no penalty if either their total waiting time exceeds a preset threshold or if they have completed a sufficient number of tasks (a single discussion in Cicero-Sync).

In addition, while our gating process is designed to select workers serious about the task, we do incorporate several techniques to assure that workers stay active when a task

gets assigned to them. Individual tasks, such as assess tasks, impose anti-cheating mechanisms to discourage spammers from quickly progressing. These mechanisms include character and word count minimums and disabling of copy-paste for free-form entries. Workers are also encouraged to peer-regulate during discussion — participants can indicate a partner's inactivity upon ending a discussion with no agreement. Paired with corresponding payout incentives, these methods ensure that most workers stay active throughout the duration of an experiment.

4 EXPERIMENTS

We deployed our experiments on our synchronous implementation, Cicero-Sync, to address the following questions: 1) Does multi-turn discussion improve individual accuracy more compared to existing one-shot reconsider based workflows? 2) Is multi-turn discussion effective in cases where acquiring justifications to implement one-shot argumentation (reconsider) is impractical? 3) Do discussions exhibit multi-turn and contextual properties?

We selected two domains to evaluate the research questions above: a traditional NLP binary answer task, Relation Extraction, for comparing against one-shot argumentation and a multi choice answer task, inspired by the word relation game Codenames, to evaluate Cicero in a non-binary choice domain.

In the following sections, we first introduce the experiment setup and configuration, then we introduce each domain and present our results for experiments on that domain. At the end, we present a qualitative analysis of discussion characteristics and explore whether discussions can improve future accuracy.

Experiment Setup

Cicero's design enables interleaved assignment of different task types (assessments or discussions) for individual workers. This can be beneficial in reducing worker waiting overhead by assigning individual tasks when paired tasks are not available. However, in order to evaluate the effects of contextual, multi-turn argumentation under a controlled setting, we need to isolate the process of assessment and argumentation. For our experiments, we implemented a "blocking" task assigner that avoids interleaved concurrent tasks and is designed to assign the same type of task to a worker until they have answered all questions of that type.

The *blocking assigner* includes a few extra constraints in addition to those required by the workflow:

- **Gold Standard Assessments**: The task assigner assigns **assess** tasks for gold standard questions to evaluate quality of workers who passed the training and gating quiz phase. Workers are assigned these questions before any other questions. No discussions are ever initiated for these questions; they let us control for worker quality and filter workers that do not pass the gating threshold.
- **Greedy Matching**: The task assigner tries to assign a discussion as soon as such a task is available. In the case of multiple candidates, the task assigner picks one randomly.

Additionally, the *blocking assigner* doesn't allocate any discussions until a worker has finished *Assess*-ing all questions. This allows us to collect the initial answers of a worker before they participate in any argumentation.

We adjusted Cicero-Sync to include these experimental constraints. The resulting system used in experiments consists of three distinct stages: *Training*, *Assess* and *Discussion / Reconsider* with workers progressing through each stage sequentially.

We conducted a between subjects experiment with 2 conditions. In the **discussion** condition, workers are matched to partners in synchronous discussion sessions after they complete the *Assess* stage according to the allocation policy described earlier. In the **reconsider** condition, we implemented the adaptive workflow and task interface as described in MicroTalk [12] to represent one-shot argumentation. In this condition, workers are adaptively asked to justify or do reconsider tasks depending on their initial answer: When a worker is the only worker with a particular answer for a question, they will be asked to provide a justification for their answer. Reconsider tasks are only assigned to a worker if there is a previously justified answer opposing their current answer. We evaluated the Relation Extraction domain with this experiment setup.

Additionally, we examined the performance of Cicero on multiple choice questions with many answers through the Codenames domain. It is infeasible to run a **reconsider** condition on this domain (as we detail later), so workers only participate in the **discussion** workflow. We also included an extra *individual assessment* stage to examine whether workers were learning from discussions. For simplicity, we may refer to this as the **codenames** condition.

Recruiting and Incentives

We ran experiments on Amazon Mechanical Turk, using workers who had completed at least 100 tasks with a 95% acceptance rate for both of our experiment domains. We recruited a total of 102 workers across the discussion, reconsider, and codenames conditions ($60, 28, 14$ respectively), with a gating pass ratio of 64%, 43%, 63% for each respective condition. Worker drop-out (post-gating) was 1, 0, 2 for each respective condition.

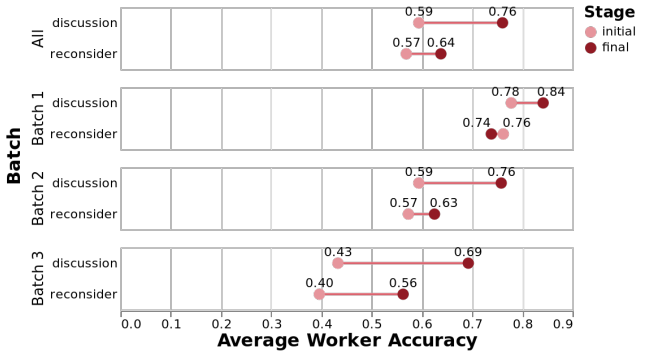

Figure 4: Comparison for improvement in average worker accuracy (Relation Extraction domain) for each batch (subset) of questions (Batches 1–3) as well as on the entire set of questions (All).

Within each domain, we calibrated our subtask payments by observing the average worker time for that subtask from a pilot run and allocating an approximately $7 hourly wage. Our training bonus of $1.00 for successfully completing training and the gating quiz is also calibrated using the average time it takes workers to complete the training session.

For the Relation Extraction domain (**discussion** and **reconsider** conditions), workers are paid $0.10 as base payment and $1.00 for passing the *Training* stage. Workers are then paid a per-question bonus of $0.05 for an assessment and $0.05 for a justification during the *Assess* stage. Depending on the condition, a bonus of $0.50 is paid for participating in a **discussion** task and $0.05 for a **reconsider** task in the last stage. Note that in the **discussion** condition, a justification is always collected for each question during the *Assess* stage so workers always get a $0.10 per-question bonus. These per-question incentives are chosen to match those used in MicroTalk [12].

For the Codenames domain, workers are paid $0.20 as base and $1.00 for passing the *training* stage. Workers are paid a per-question bonus of $0.20 for each correct answer and a per-discussion bonus of $0.50 for participating in a discussion with an extra $0.25 for holding the correct answer at the end of discussion.

While it is possible to design a more complex incentive structure, our main goal for this set of incentives is to discourage cheating behavior and align with that of MicroTalk. We think these incentives are consistent with those used in other, recent crowdsourcing research [27].

Relation Extraction Domain: Binary Answer

In the interest of comparing to previous work, we evaluated our method on a tradition NLP annotation task of *information extraction* (IE) — identifying structured, semantic information (relational tuples, such as would be found in a SQL database) from natural language text [13]. The task is of considerable interest in the NLP community, since most IE approaches use machine learning and many exploit crowdsourced training data [1, 27, 33, 46].

Specifically, we consider the problem of annotating a sentence to indicate whether it encodes the TAC KBP *LivedIn* relation — does a sentence support the conclusion that a person lived in a location? While such a judgment may seem simple, the official LDC annotation guidelines are deceptively complex [40]. For example, one can conclude that a national official lives in her country, but not that a city official lives in her city. Figure 3 defines the task, showing the instructions given to our workers.

We created a set of 23 challenging TAC KBP questions drawing from the 20 used in MicroTalk [12] and adding 3 additional questions from Liu *et al.* [27]. This set was then divided into 3 batches of size 7, 8, and 8 for our discussion experiments. For gold standard questions, we selected 3 simple questions from the TAC KBP set, each of which can be resolved with an invocation of one rule. Upon recruitment, each worker is also presented with a 6 question gating quiz and are allowed 2 attempts to pass the gating threshold. Gating questions were written to be simple and unambiguous, testing whether the worker was diligent and had absorbed the guidelines.

Multi-turn *vs.* One-shot Workflows

Our first experiment compares worker accuracy for the multi-turn, contextual discussion workflow design against that of a one-shot (non-contextual) reconsider workflow on the binary answer Relation Extraction domain (*i.e.*, Cicero *vs.* MicroTalk). We deployed both conditions with the configuration described in the experiment setup with the gating threshold set at 100% — workers needed to answer all gold standard questions correctly to be included. Also, since workers need to complete all assessments before starting discussions which would cause increased waiting time on a large set of questions, we deployed the *discussion* condition experiments in 3 batches (N=9, 16, 13) corresponding to the 3 batches the experiment questions were divided into. In the *reconsider* condition (N = 12), workers were put through our implementation of the adaptive workflow from MicroTalk on all questions.

From the plot shown in Figure 4 we can see that the *discussion* condition (Cicero) improves average worker accuracy by 16.7 percentage points over the initial accuracy compared to 6.8 for the *reconsider* condition (statistically significant, t-test at $p = 0.0143$).

We performed a t-test on the initial accuracy of workers across both conditions for each batch and found no statistically significant difference ($p = 0.77, 0.78, 0.67$) indicating

Candidates	business, card, knot
Positive Clues	suit, tie
Negative Clues	corporation, speed
Explanation	Workers must find the single best candidate word that is related in meaning to some positive clue word, but none of the negative clues. In this example, all three candidates are related to some positive clue: a suit for business, a suit of cards, and to tie a knot. However, business relates to corporation and knot is a unit of speed. Card is the best answer: it's related to a positive clue, while being largely unrelated to any negative clues.
Best Answer	card

Table 1: Example of a simple question used for training from the Codenames domain. Real questions have around 7-10 candidate words.

that workers of similar quality were recruited for each of our batches. On average, workers participated in 7.7 discussions ($\sigma = 4.75$) and were presented with 16.8 reconsider prompts ($\sigma = 3.83$) in the one-shot workflow.

We do note that discussions are more costly, largely due to paying workers for time spent waiting for their partner to respond. Each Cicero-Sync discussion took an average of 225.3 seconds ($\sigma = 234.8$) of worker time compared to a one-shot reconsider task averaging 13.6 seconds ($\sigma = 15.0$). We believe that an asynchronous implementation of Cicero could reduce overhead and dramatically lower costs.

Codenames Domain: Multiple Choice with Many Answers

Previous work using one-shot argumentation [12, 37] focused mainly on evaluating argumentation in domains that only acquired binary answers such as Relation Extraction or sarcasm detection. These systems ask workers to fully justify their answer, which can be done by arguing against the opposing answer and for one's own.

However, we observed that this is not sufficient to represent a wide variety of real world tasks. As the number of answer options grows, it becomes increasingly inefficient and even infeasible to ask workers to provide full, well-argued justification for their answers beforehand. Full justifications for multiple choice answers would need to address not only the worker's own answer, but also argue against *all* remaining options, making the justifications long and difficult to understand. Multi-turn discussion can address these scaling issues through back-and-forth dialog through which workers argue only against their partner's specific answer.

Inspired by the popular word association *Codenames* board game, we created a new test domain that requires choosing between numerous possible answers. Similar game-based domains have been adopted to evaluate cooperative work designs such as in DreamTeam [47], which utilized a cooperative version of Codenames, and CrowdIA [24], which used a mystery game. The objective in the game is for each team to identify the tiles assigned to them from a shared list of word tiles. Clue words are given by one team member (the "spymaster") who can see the assignment of word tiles (which ones belong to which team) while other teammates have to find the correct word tiles for their team while avoiding the tiles assigned to the other team.

Our Codenames task domain draws inspiration from the competitive aspect of the game. We observe that late into the game, good word guesses are often informed by both the teammate clues (which should be matched) and opponent clues (which should be avoided). With this observation, we created tasks which consist of a list of candidate words, several positive and several negative clue words. Workers, in the role of a team member, are instructed to find the single best candidate word that is related in meaning to some positive clue word but none of the negative clues. An example of this task can be seen in Table 1. Each question contains around 2 positive clues, 2–3 negative clues and 7–10 candidate words. We created 3 gating questions, 7 experiment questions, and 1 question for the *individual assessment* stage for this task. We used a gating threshold of 66.7%. While Codenames is not a typical task for crowd work, as also noted in DreamTeam, we think its aspect of multiple choice answers is representative of a whole class of similar tasks that lack effective one-shot argumentation strategies.

The loose definition of words being "related" in the Codenames domain reduces the amount of worker training required for participation since it utilizes common knowledge of language. However, this may lead to ambiguity in reference answers which would be undesirable. We elected to manually create a set of questions which were validated to have only 1 objectively best answer. The distractors for each question and our reference argument were evaluated with a group of expert pilot testers. We confirmed that all participants agreed with our reference counter-arguments against the distractors and also with our reference answer. In the pilot test, we also noted that this task can be very challenging even for experts as multiple word senses are involved in distractors.

Evaluating on Multiple Choice Tasks with Many Answers

Our second experiment (N = 12) examines the performance of Cicero-Sync on multiple choice answer tasks from the Codenames domain, a domain that would be very inefficient

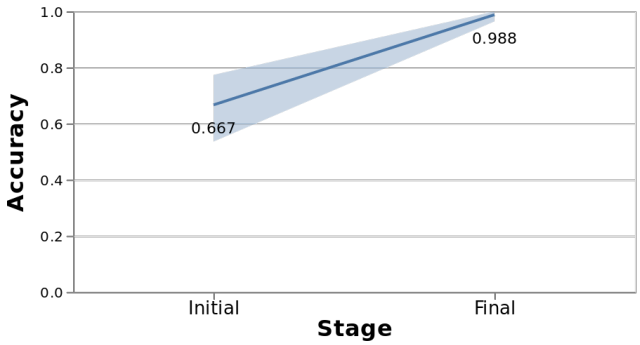

Figure 5: Initial and final accuracy of multi-turn argumentation on the Codenames domain with 95% confidence intervals.

for one-shot argumentation (justifications would need to address up to 9 alternatives). We achieved a final average worker accuracy of 98.8% compared to 66.7% initial accuracy (Figure 5) – a 32.1 percentage point improvement.

We tested the significance of this improvement through an ANOVA omnibus test with a mixed effects model using worker initial accuracy as a random effect and found that the improvement was statistically significant at ($F(1, 57.586) = 85.608, p = 5.445x10^{-13} < 0.001$). The average duration of each discussion was 123.56 seconds ($\sigma = 64.79$) and each worker had an average of 6.3 discussions ($\sigma = 3.89$).

Discussion Characteristics

We can see from the previous experiments that multi-turn, contextual argumentation is effective at improving worker accuracy across a variety of tasks, but are the discussions actually taking advantage of multi-turn arguments and the context being available? To answer this question, we collected and analyzed the transcripts recorded for each domain: Relation Extraction and Codenames.

We computed statistics on multi-turn engagement by analyzing the number of worker-initiated messages – each of which is considered a turn. We found that in the Relation Extraction domain, discussions averaged 7.5 turns ($\sigma = 6.1$, median of 5) while in the Codenames domain discussions averaged 8.3 turns ($\sigma = 4.23$, median of 7). We also found that in Codenames, the number of turns correlates to convergence on the correct answer ($F(1, 31) = 7.2509, p < 0.05$) while we found no significant relation between turns and convergence ($p > 0.1$) in the Relation Extraction domain. We note that in Relation Extraction, discussions are seeded with workers' justifications from the assess task (equivalent to 2 non-contextual turns, which should be added to the average numbers above for comparison purposes) whereas discussions in the Codenames domain use actual contextual turns to communicate this information. Compared to workers in Relation Extraction conditions, workers in the Codenames discussions sometimes utilized extra turns to reason about alternative choices neither worker picked when entering the discussion.

	Relation Extraction	Codenames
Refute	42%	59%
Query	25%	35%
Counter	34%	14%
Previous	16%	10%

Table 2: Proportion of each pattern appearing in discussions that converged to the correct answer for each domain. Refute and Query suggest utility of multi-turn interactions while Counter and Previous mainly suggest utility of context.

Additionally, we noticed several patterns in the discussion text that appeared in both domains. We further examined these patterns by coding the the discussion transcripts (147 from Relation Extraction and 38 from Codenames). We surveyed the discussions looking only at patterns specific to argumentation and came up with 8 patterns related to argumentation techniques and 6 reasons workers changed their answer.

We then narrowed down the argumentation patterns by removing any that were highly correlated or any that had just 1–2 examples and finalized the following 4 prominent patterns as codes:

- **Refute**: Argue by directly giving a reason for why the partner's specific *answer* is believed to be incorrect. Examples: "Small [partner choice] is the opposite of large [negative clue] and will not work"; "Louisana [sic] isn't a country, therefore NonCountry applies."
- **Query**: Ask the partner to explain their answer, a part of their answer or ask for a clarification in their explanation. Examples: "Why do you think it should be bill?"; "How would bridge work?"
- **Counter**: Pose a counter-argument to a partner *in response to* their explanation. Example: A: "Erdogan's government is nationally affiliated with Turkey." B: "[...] The sentence could be interpreted as one of Turkey's allies is helping them with the EU thing."
- **Previous**: Explicitly state that knowledge/line of reasoning acquired from a previous discussion is being used. Example: "I had window at first too, but someone else had bridge, but they thought bridge because of the card game bridge, and that made sense to me";

We found that workers used these contextual patterns frequently during their discussions for both domains with

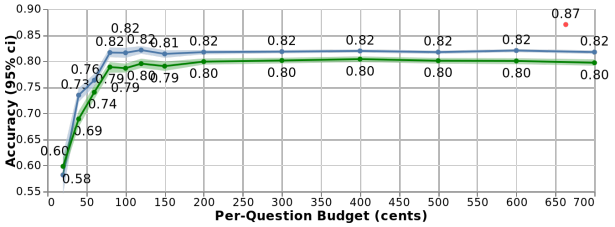

Figure 6: Scaling of majority vote (green) and EM-aggregated performance (blue) for one-shot argumentation (Microtalk) on the Relation Extraction domain, computed by simulation (100 simulations per budget) excluding training cost. While expensive due to the use of real-time crowdsourcing, EM-aggregated performance of Cicero-Sync **(shown as a red dot) is higher.**

77.6% and 86.8% of all discussions utilizing at least one pattern in the Relation Extraction and Codenames domains respectively. We can also see that distribution of patterns across the two domains (Table 2) on discussions converging to the correct answer indicates that the utility of each pattern may be different in different domains. We hypothesize that the higher frequency of **Counter** and lower frequency of **Query** in Relation Extraction is likely due to the justification seeding which reduced need for workers to ask for explanations but encouraged more counter-arguments.

We also condensed the reasons for workers changing their answer down to 3 basic categories: learning about the *task* (rules), agreeing on meaning of concepts in a *question*, and being *convinced* by an argument. After coding the discussions, we found that the distribution of the reason for changing answers was 18%, 3%, 79% for Relation Extraction domain and 17%, 28%, 55% for Codenames, across each category (task, question, convinced) respectively showing that discussions could help workers understand the task.

We also observed that 70% of all discussions and 75% of discussions converging to the right answer used our rule shorthands when referring to the rules instead of describing them. However, we note that simply citing shorthands doesn't correlate with convergence of a discussion ($p > 0.1$).

Do Workers Learn Through Discussion?

While we didn't design discussions to be used as a way of training workers, many reported that they "understood the task much better" after discussions in pilot experiment feedback so we explored the effects of discussions on workers' future accuracy. We tested a worker's performance by adding post-test questions after they finished their corresponding experiment condition. We selected 4 questions for the Relation Extraction domain and 1 for the Codenames domain, all of comparable difficulty to the main questions, to be individually evaluated.

Average accuracy on the individual evaluation sections trended higher for the discussion condition: accuracies were 66.7%, 69.3%, and 73.9% for the *baseline* (no argumentation), *reconsider* and *discussion* conditions respectively in the Relation Extraction domain and 46.7% and 52.0% for the *baseline* and *discussion* conditions in the Codenames domain. However, ANOVA on all conditions for each domain shows no statistically significant interaction ($F(1, 49.1) = 0.013, p > 0.1$ and $F(2, 58.3) = 1.03, p > 0.1$ for Codenames and Relation Extraction respectively) between the experiment condition and the accuracy on the individual evaluation questions. We conjecture that need for argumentation may be reduced as workers better learn the guidelines through peer interaction [11, 23], but the difficult questions will likely always warrant some debate.

5 DISCUSSION

While each **discussion** task in Cicero-Sync required more worker time, we found significantly higher gains to individual worker accuracy compared to the **reconsider** condition from MicroTalk. We believe that much of the increase in work time stems from our decision to use synchronous, real-time crowdsourcing in Cicero-Sync, leading to higher per-argument-task costs. Under a synchronous environment, workers must wait for other workers' actions during and in-between discussions. Since our experiments are focused on *evaluating* the multi-turn argumentation workflow, synchronized discussions allowed us to better collect data in a controlled way. Many efficiency optimizations, that we did not explore, could be implemented to run the Cicero workflow at scale in a more cost effective way. Specifically, an asynchronous implementation of Cicero would eliminate the need for workers to wait for each other, reducing costs. However, if the synchronous implementation were run at larger scale on a much larger set of problems, there would be proportionately less overhead. A semi-asynchronous workflow can be created using notifications and reminder emails [37]. Larger asynchronous group discussions can also be made possible through summarizing discussions [45] thus reducing the cost of new participants getting up to speed.

Argumentation, whether one-shot or multi-turn, may not be appropriate for many tasks, even those requiring high-effort [5]. For example, if one is merely labeling training data for supervised machine learning (a common application), then it may be more cost effective to eschew most forms of quality control (majority vote, EM or argumentation) and instead collect a larger amount of noisy data [25]. However, if one needs data of the highest possible accuracy, then argumentation — specifically contextual, multi-turn argumentation — is the best option. We simulated the effects of recruiting more workers according to the policy described in [12] at higher budgets. Figure 6 shows performance for one-shot

argumentation after aggregating answers across all workers using EM along with the aggregated Cicero-Sync results. We observe that accuracy plateaus for one-shot argumentation, confirming previous reports [9, 12], and that Cicero achieves 5 percentage points higher aggregated accuracy compared to previous work, even when accounting for the higher cost of multi-turn discussions.

In the end, the most cost effective crowd technique depends on both problem difficulty and quality requirements. High-cost methods, like argumentation, should be reserved for the most difficult tasks, such as developing challenging machine learning *test* sets, or tasks comprising a high-stakes decision, where a corresponding explanation is desirable.

6 CONCLUSION & FUTURE WORK

In this paper, we explored the potential for multi-turn, contextual argumentation as a next step for improving crowdsourcing accuracy. We presented Cicero, a novel workflow that engages workers in *multi-turn, contextual* argumentation (discussion) to improve crowd accuracy on difficult tasks. We implemented this workflow using a synchronous, real-time design for discussions tasks and created the Cicero-Sync system. Since the quality of a discussion depends on its participants, we also designed and implemented gated instructions and a novel justification training task for Cicero-Sync to ensure competent discussions through improving workers' ability to recognize and synthesize good arguments.

We demonstrate that our implementation of Cicero-Sync, the synchronous version of the Cicero workflow, is able to achieve two things:

- Higher improvement in accuracy compared to a state-of-art, one-shot argumentation system on a difficult NLP annotation task: a 16.7 percentage point improvement over individual workers' pre-argumentation accuracy *v.s.* a 6.8 point improvement using one-shot argumentation and 5 percentage points higher aggregate accuracy when aggregating the opinions of multiple workers using majority vote or EM, accounting for cost.
- Very high accuracy in a non-binary choice answer task that would be impractical with one-shot argumentation: 98.8% accuracy (a 32.1 percentage point improvement over the initial accuracy.)

Both these accuracies are much higher than can be achieved without argumentation. Traditional majority vote and EM without argumentation approaches plateau at 65% on similar questions [12]. Additionally, we observed several interesting patterns of discourse that are enabled by multi-turn, contextual argumentation and note that many successful discussions utilize these patterns.

There are many future directions for improving the argumentation workflow and system implementation. Currently, the cost of argumentation is still relatively high but cost may be reduced further as discussed earlier.

There are also details in the interactions that could be examined in future work. While we kept workers anonymous between discussions, benefits of assigning pseudonyms as a persistent identity [37] in repeated sessions may be worth considering. Additionally, the idea of utilizing worker produced highlights to refer to the task guidelines and question in [37] could be incorporated in a future iteration to extend our concept of rule shorthands.

We also envision that better models of discussions could allow a future system to only pair arguments where the outcome reduces uncertainty. Furthermore, there is potential in incorporating natural language processing techniques to identify and support positive behavior patterns during argumentation and opportunities for learning from misconceptions surfaced during discussion to improve training and task instructions [4].

Finally, we believe argumentation techniques can be extended to a wider range of tasks and meta-tasks, including issues like micro-task organization studied in Turkomatic [22] and flash teams [35], as well as offer new avenues for human-machine collaboration.

7 ACKNOWLEDGEMENTS

We would like to thank Eunice Jun, Gagan Bansal and Tongshuang Wu for their helpful feedback and participation in our pilot experiments as well as the anonymous reviewers for their helpful comments. This work was supported in part by NSF grant IIS-1420667, ONR grants N00014-15-1-2774 and N00014-18-1-2193, the WRF/Cable Professorship and support from Google.

REFERENCES

[1] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D. Manning. 2014. Combining Distant and Partial Supervision for Relation Extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1556–1567. http://aclweb.org/anthology/D/D14/D14-1164.pdf

[2] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *UIST '10 Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM Press, 313–322.

[3] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. 2010. VizWiz: Nearly Real-time Answers to Visual Questions. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10)*. ACM, New York, NY, USA, Article 24, 2 pages. DOI:http://dx.doi.org/10.1145/1805986.1806020

[4] Jonathan Bragg, Mausam, and Daniel S. Weld. 2018. Sprout: Crowd-Powered Task Design for Crowdsourcing. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 165–176. DOI:http://dx.doi.org/10.1145/3242587.3242598

[5] Justin Cheng, Jaime Teevan, and Michael S. Bernstein. 2015. Measuring Crowdsourcing Effort with Error-Time Curves. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1365–1374. DOI: http://dx.doi.org/10.1145/2702123.2702145

[6] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. 2013. Cascade: Crowdsourcing Taxonomy Creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1999–2008. DOI: http://dx.doi.org/10.1145/2470654.2466265

[7] Peng Dai, Christopher H. Lin, Mausam, and Daniel S. Weld. 2013. POMDP-based control of workflows for crowdsourcing. *Artificial Intelligence* 202 (2013), 52–85.

[8] A.P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-rates using the EM Algorithm. *Applied Statistics* 28, 1 (1979), 20–28.

[9] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 469–478. DOI:http://dx.doi.org/10.1145/2187836.2187900

[10] Jeff Donahue and Kristen Grauman. 2011. Annotator rationales for visual recognition. *2011 International Conference on Computer Vision* (2011), 1395–1402.

[11] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the Crowd Yields Better Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1013–1022. DOI:http://dx.doi.org/10.1145/2145204.2145355

[12] Ryan Drapeau, Lydia B Chilton, Jonathan Bragg, and Daniel S Weld. 2016. Microtalk: Using argumentation to improve crowdsourcing accuracy. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

[13] Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *Information extraction a multidisciplinary approach to an emerging information technology*. Springer, 10–27.

[14] Chien-Ju Ho and Ming Yin. 2018. Working in Pairs: Understanding the Effects of Worker Interactions in Crowdwork. *Computing Research Repository, CoRR* abs/1810.09634 (2018).

[15] Ting-Hao Kenneth Huang and Jeffrey P Bigham. 2017. A 10-Month-Long Deployment Study of On-Demand Recruiting for Low-Latency Crowdsourcing. In *Proceedings of The fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2017)*.

[16] Ting-Hao Kenneth Huang, Walter S Lasecki, Amos Azaria, and Jeffrey P Bigham. 2016. " Is There Anything Else I Can Help You With?" Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

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

[18] Ece Kamar, Severin Hacker, and Eric Horvitz. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 467–474.

[19] David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Budget-optimal Crowdsourcing using Low-rank Matrix Approximations. In *Conference on Communication, Control, and Computing*.

[20] Masaki Kobayashi, Hiromi Morita, Masaki Matsubara, Nobuyuki Shimizu, and Atsuyuki Morishima. 2018. An Empirical Study on Short- and Long-Term Effects of Self-Correction in Crowdsourced Microtasks. In *AAAI Conference on Human Computation and Crowdsourcing, HCOMP*.

[21] Travis Kriplean, Jonathan T. Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2011. ConsiderIt: Improving Structured Public Deliberation. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 1831–1836. DOI:http://dx.doi.org/10.1145/1979742.1979869

[22] Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with Turkomatic. In *CSCW*. ACM Press, New York, New York, USA. DOI:http://dx.doi.org/10.1145/2145204.2145354

[23] Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R Klemmer. 2015. Peer and self assessment in massive online classes. In *Design thinking research*. Springer, 131–168.

[24] Tianyi Li, Kurt Luther, and Chris North. 2018. CrowdIA: Solving Mysteries with Crowdsourced Sensemaking. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 105 (Nov. 2018), Article 105, 29 pages. DOI: http://dx.doi.org/10.1145/3274374

[25] Christopher H. Lin, Mausam, and Daniel S. Weld. 2014. To Re(label), or Not To Re(label). In *HCOMP*.

[26] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2009. TurKit: Tools for Iterative Tasks on Mechanical Turk. In *Human Computation Workshop (HComp2009)*.

[27] Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H Lin, Xiao Ling, and Daniel S Weld. 2016. Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 897–906.

[28] Andrew Mao, Yiling Chen, Krzysztof Z Gajos, David Parkes, Ariel D Procaccia, and Haoqi Zhang. 2012. Turkserver: Enabling synchronous and longitudinal online experiments. *Proceedings of HCOMP* 12 (2012).

[29] Tyler McDonnell, Matthew Lease, Tamer Elsayad, and Mucahid Kutlu. 2016. Why Is That Relevant? Collecting Annotator Rationales for Relevance Judgments. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 10.

[30] Sarah Michaels, Catherine O'Connor, and Lauren B Resnick. 2008. Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in philosophy and education* 27, 4 (2008), 283–297.

[31] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z. Gajos. 2011. Platemate: Crowdsourcing Nutritional Analysis from Food Photographs. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 1–12. DOI:http://dx.doi.org/10.1145/2047196.2047198

[32] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing.. In *Human Computation Workshop*. 11.

[33] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. 2014. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of ACL*.

[34] Drazen Prelec and H. Sebastian Seung. 2007. An algorithm that finds truth even if most people are wrong. (2007). Working Paper.

[35] Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S Bernstein. 2014. Expert crowdsourcing with flash teams. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 75–85.

[36] Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (Nov. 2018), Article 151, 26 pages. DOI: http://dx.doi.org/10.1145/3274420

[37] Mike Schaekermann, Joslin Goh, Kate Larson, and Edith Law. 2018. Resolvable vs. Irresolvable Disagreement: A Study on Worker Deliberation in Crowd Work. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 154 (Nov. 2018), Article 154, 19 pages. DOI:http://dx.doi.org/10.1145/3274423

[38] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and A. Ng. 2008. Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *2008 Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[39] Yu-An Sun, Christopher R Dance, Shourya Roy, and Greg Little. 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*.

[40] Mihai Surdeanu. 2013. Overview of the TAC2013 Knowledge Base Population Evaluation: English Slot Filling and Temporal Slot Filling. In *TAC 2013*.

[41] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *Advances in neural information processing systems (NIPS)*. 2424–2432.

[42] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Laberlers of Unknown Expertise. In *In Proc. of NIPS*. 2035–2043.

[43] J. Wiebe, R. Bruce, and T. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. 246–253. http://dl.acm.org/citation.cfm?id=1034678.1034721

[44] Omar F Zaidan, Jason Eisner, and Christine D Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Proceedings of NAACL and HLT 2007*.

[45] Amy X. Zhang and Justin Cranshaw. 2018. Making Sense of Group Chat Through Collaborative Tagging and Summarization. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 196 (Nov. 2018), Article 196, 27 pages. DOI:http://dx.doi.org/10.1145/3274465

[46] Ce Zhang, Feng Niu, Christopher Ré, and Jude Shavlik. 2012. Big data versus the crowd: Looking for relationships in all the right places. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 825–834.

[47] Sharon Zhou, Melissa Valentine, and Michael S. Bernstein. 2018. In Search of the Dream Team: Temporally Constrained Multi-Armed Bandits for Identifying Effective Team Structures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 108, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3173682

[48] Haiyi Zhu, Steven P. Dow, Robert E. Kraut, and Aniket Kittur. 2014. Reviewing Versus Doing: Learning and Performance in Crowd Assessment. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 1445–1455. DOI:http://dx.doi.org/10.1145/2531602.2531718