The Dynamics of Micro-Task Crowdsourcing

The Case of Amazon MTurk

Djellel Eddine Difallah*, Michele Catasta†, Gianluca Demartini‡,
Panagiotis G. Ipeirotis◇, Philippe Cudré-Mauroux*
* eXascale Infolab, University of Fribourg, Switzerland
† EPFL, Switzerland
‡ University of Sheffield, UK
◇ New York University, USA

ABSTRACT

Micro-task crowdsourcing is rapidly gaining popularity among research communities and businesses as a means to leverage Human Computation in their daily operations. Unlike any other service, a crowdsourcing platform is in fact a marketplace subject to human factors that affect its performance, both in terms of speed and quality. Indeed, such factors shape the *dynamics* of the crowdsourcing market. For example, a known behavior of such markets is that increasing the reward of a set of tasks would lead to faster results. However, it is still unclear how different dimensions interact with each other: reward, task type, market competition, requester reputation, etc.

In this paper, we adopt a data-driven approach to (A) perform a long-term analysis of a popular micro-task crowdsourcing platform and understand the evolution of its main actors (workers, requesters, tasks, and platform). (B) We leverage the main findings of our five year log analysis to propose features used in a predictive model aiming at determining the expected performance of any batch at a specific point in time. We show that the number of tasks left in a batch and how recent the batch is are two key features of the prediction. (C) Finally, we conduct an analysis of the demand (new tasks posted by the requesters) and supply (number of tasks completed by the workforce) and show how they affect task prices on the marketplace.

Categories and Subject Descriptors

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

General Terms

Design; Experimentation; Human Factors

Keywords

Crowdsourcing; trend identification; tracking; forecasting

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741685.

1. INTRODUCTION

While general data availability increases, its quality is not necessarily perfect and manual data pre-processing is often necessary before using it to create value or to support decisions. To this end, outsourcing data-processing tasks like, for example, image tagging, audio transcription, translation, etc. to a large crowd of individuals on the Web has become more popular over time.

To perform such Human Intelligence Tasks (HITs), crowdsourcing platforms have been developed. Such platforms serve as a place where the crowd (*workers*) willing to perform small tasks (so called *micro-tasks*) in exchange of a small monetary reward and work providers (also known as *requesters*) meet.

The micro-task crowdsourcing market has seen a rapid growth in the last five years. This is also explained by the fact that large amounts of data are today available in companies, which are increasingly seen as a key asset for optimizing all business processes.

The micro-task crowdsourcing process works as follows. First, the requesters design the HIT based on their data and required task. Next, they publish batches of HITs on the crowdsourcing platform specifying their requirements and the monetary amount rewarded to workers in exchange of the completion of each HIT. Then, the workers willing to perform the published HITs complete the tasks and submit their work back to the requester who obtains the desired results and pays workers accordingly.

In this paper, we analyze the evolution of a very popular micro-task crowdsourcing platform (i.e., Amazon MTurk[1]) over a five-year time span and report key findings about how the market behaves with regards to demand and supply. Using features derived from a large-scale analysis of observations made on the platform, we propose methods to predict the throughput of the crowdsourcing platform for a batch of HITs published by a given requester at a certain point in time. This prediction is based on different features including the current platform load and the task type. Using this prediction method, we try to understand the impact of each feature that we consider, and its scope over time.

The main findings of our analysis are: 1) the type of tasks published on the platform has changed over time with content creation HITs being the most popular today; 2) the HIT pricing approach evolved towards larger and higher paid HITs to better attract workers in a competitive market; 3) geographical restrictions are applied to certain task types

[1] `http://mturk.com`

(e.g., surveys for US workers); 4) we observe an organic growth in the number of new requesters who use the platform, which is a sign of a healthy market; 5) we identify *size of the batch* as the main feature that impacts the progress of a given batch; 6) we observe that supply (the workforce) has little control over driving the price of demand.

In summary, the main contributions of this paper are:

- An analysis of the evolution of a popular micro-task crowdsourcing platform looking at dimensions like topics, reward, worker location, task types, and platform throughput.
- A large-scale classification of 2.5M HIT types published on Amazon MTurk.
- A predictive analysis of HIT batch progress using more than 29 different features.
- An analysis of the crowdsourcing platform as a market (demand and supply).

The rest of the paper is structured as follows. In Section 2, we overview recent work on micro-task crowdsourcing specifically focusing on how micro-task crowdsourcing has been used and on how it can be improved. Section 3 presents how Amazon MTurk has evolved over time in terms of topics, reward, and requesters. Section 4 summarizes the results of a large-scale analysis on the types of HIT that have been requested and completed over time. Based on the previous findings, Section 5 presents our approach to predicting the throughput of the crowdsourcing platform for a batch of published HITs. Section 6 studies the Amazon MTurk market and how different events correlate (e.g., new HITs attracting more workers to the platform). We discuss our main findings in Section 7 before concluding in Section 8.

2. RELATED WORK

The objective of this piece of work is to understand and characterize how a micro-task crowdsourcing platform behaves as a marketplace. Thus, we first start by reviewing related work on human computation and micro-task crowdsourcing, before turning to related work on market analysis and on how to improve crowdsourcing platforms.

Micro-task Crowdsourcing.

Crowdsourcing is defined as the outsourcing of tasks to a crowd of individuals over the Web. Crowdsourcing has been used for a variety of purposes, from innovation to software development [23]. Early crowdsourcing examples leveraged the fun or community belonging incentives (e.g., Wikipedia) instead of monetary rewards. Examples of crowdsourcing systems based on gamification include the ESP game [21] where players must agree on tags to use for a picture without the possibility to interact with each other. An extension of the ESP game is Peekaboom: a game that asks players to annotate specific objects within an image [22].

In this work, we focus on *paid micro-task crowdsourcing*, where the crowd is asked to perform short tasks, also known as Human Intelligence Tasks (HITs), in exchange for a small monetary reward per unit. Popular examples of such tasks include: spell checking of short paragraphs, sentiment analysis of tweets, rewriting product reviews, or transcription of scanned shopping receipts.

Micro-task crowdsourcing is often used to improve the quality of machine-run algorithms in order to combine both the scalability of machines over large amounts of data as well as the quality of human intelligence in processing and understanding data [21]. Many examples of such hybrid human-machine approaches exist. Crowd-powered databases [9] leverage crowdsourcing to deal with problems like data incompleteness, data integration, graph search, and joins [24, 18, 25]. Semantic Web systems leverage the crowd for tasks like schema matching [19], entity linking [4], and ontology engineering [17]. Information Retrieval systems have used crowdsourcing for evaluation purposes [1]. Models and paradigm for hybrid human-machine systems have been proposed on top of crowdsourcing platforms [15], also including the design of hybrid workflows [16].

In this work, we specifically focus on micro-task crowdsourcing and analyze the dynamics of a very popular platform for this purpose: Amazon MTurk. This platform provides access to a crowd of workers distributed worldwide but mainly composed of people based in the US and India [12]. Many Amazon MTurk workers share their experience about HITs and requesters through dedicated web forums and ad-hoc websites [13]. Requester 'reviews' serve as a way to measure the reputation of the requesters among workers and it is assumed to influence the latency of the tasks published [20], as workers are naturally more attracted by HITs published by requesters with a good reputation.

Because of the complex mechanisms connecting the workers to the requesters and to the platform itself, characterizing the dynamics and evolution of micro-task crowdsourcing platforms is key in order to understand the impact of the various components and to design better human computation systems. The goal of our work is to understand the evolution of a micro-task crowdsourcing platform over time and to identify key properties of such platform.

Market Analysis.

An initial work analyzing the Amazon MTurk market was done in [12]. Our paper extends this work by considering the time dimension and analyzing long term trends. Faradani et al. [7] proposed a model to predict the completion time of a batch. Our prediction endeavor is however different, in the sense that we aim at predicting the immediate throughput based on current market conditions and to understand what features have more impact than others.

Improving Crowdsourcing Platforms.

In [14] authors give their view on how the crowdsourcing market should evolve in the future, specifically focusing on how to support full-time crowd workers. Likewise, our goal is to identify ways of improving crowdsourcing marketplaces by understanding the dynamics of such platforms—based on historical data and models.

Recent contributions on novel crowdsourcing platforms have proposed methods for identifying the best workers in the crowd for specific tasks [6, 2]. Given the diverse task types being published on micro-task crowdsourcing platforms, such functionalities could be used to improve the work experience of the crowd and the quality of the results returned to the requesters.

One way to improve the efficiency of the crowdsourcing process is to use custom HIT *pricing schemes*. For example, in [11] authors propose models to set the HIT reward given some latency and budget constraints. In [5], we studied how worker retention can improve the latency of a batch by leveraging varying bonus schemes.

Our work is complementary to existing work as we present a data-driven study of the evolution of micro-task crowdsourcing over five years. Our work can be also used to support requesters in publishing HITs on these platforms and getting results more rapidly.

3. THE EVOLUTION OF AMAZON MTURK FROM 2009 TO 2014

In this section, we start by describing our dataset and extract some key information and statistics that we will use in the rest of the paper.

3.1 Crowdsourcing Platform Dataset

Over the past five years, we have periodically collected data about HITs published on Amazon MTurk. The data that we collect from the platform is available at `http://mturk-tracker.com/`.

In this work, we consider hourly aggregated data that includes the available HIT batches and their metadata (title, description, rewards, required qualifications, etc.), in addition to their progress over time, that is, the temporal variation of the set of HITs available. In fact, one of the main metrics that we leverage (see Section 5) is the throughput of a batch, i.e., how many HITs get completed between two successive observations. In Figure 1, we plot the number of HITs available in a given batch versus its throughput. An interesting observation that can be made is that large batches can achieve high throughput (thousands of HITs per minute).

In total, our dataset covers more than 2.5M different batches with over 130M HITs. We note that the tracker reports data periodically only and does not reflect fine-grained information (e.g., real-time variations). We believe however that it captures enough information to perform meaningful, long-term trend analyses and to understand the dynamics and interactions within the crowdsourcing platform.

3.2 A Data-driven Analysis of Platform Evolution

First, we identify trends obtained from aggregated information over time, keywords, and countries associated to the published HITs. Each of the following analyses is also available as an interactive visualization over the historical data on `http://xi-lab.github.io/mturk-mrkt/`.

Topics Over Time.

First, we want to understand how different topics have been addressed by means of micro-task crowdsourcing over time. In order to run this analysis, we look at the keywords associated with published HITs. We observe the evolution of keyword popularity and associated reward on Amazon MTurk. Figure 2 shows this behavior. Each point in the plot represents a keyword associated to the HITs with its frequency (i.e., number of HITs with this keyword) on the x-axis, and the average reward in a given year on the y-axis. The path connecting data points indicates the time evolution, starting in 2009, with one point representing the keyword usage over one year.

We observe that the frequency of the 'audio' and 'transcription' keywords (i.e., blue and red paths from left to right) have substantially increased over time. They have become the most popular keywords in the last two years and are paid more than $1 on average. HITs with the 'video' tag have also increased in number with a reward that has reached a peak in 2012 and decreased after that. HITs tagged as 'categorization' have been paid consistently in the range of $0.10-$0.30 on average, except in 2009 where they were rewarded less than $0.10 each. HITs tagged as 'tweet' have not increased in number but have been paid more over the years, reaching $0.90 on average in 2014: This can be explained by more complex tasks being offered to workers, such as sentiment classification or writing of tweets.

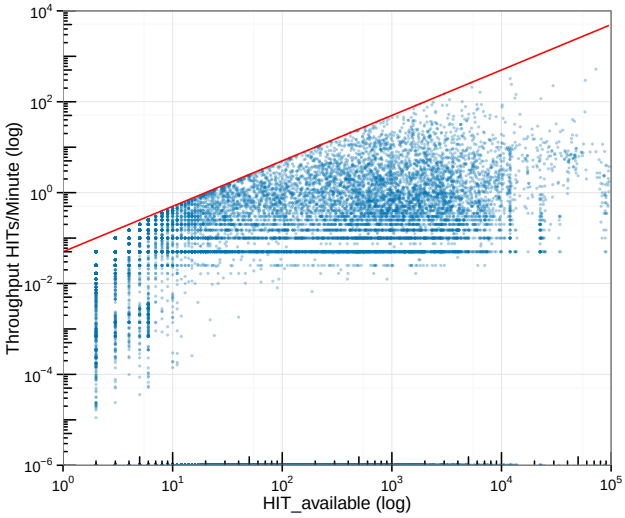


Figure 1: Batch throughput versus number of HITs available in the batch. The red line corresponds to the maximum throughput we could have observed due to the tracker periodicity constraints. For readability, this graph represents a subset of 3 months (January-March 2014), and HITs with rewards $0.05 and less.

Preferred Countries by Requesters Over Time.

Figure 3 shows the requirements set by requesters with respect to the countries they wish to select workers from. The left part of Figure 3 shows that most HITs are to be completed exclusively by workers located in the US, India, or Canada. The right part of Figure 3 shows the evolution over time of the country requirement phenomenon. The plot shows the number of HITs with a certain country requirement (on the y-axis) and its time evolution (on the x-axis) with yearly steps. The size of the data points indicates the total reward associated to those HITs.

We observe that US-only HITs dominate, both in terms of their large number as well as in terms of the reward associated to them. Interestingly, we notice how HITs for workers based in India have been decreasing over time. On the other hand, HITs for workers based in Canada have been increasing over time, becoming in 2014 larger than those exclusively available to workers based in India. We also see that the reward associated to them is smaller than the budget for India-only HITs. As of 2014, both HITs for workers based in Canada or UK are more numerous that those for workers based in India. Overall, 88.5% of the HIT batches

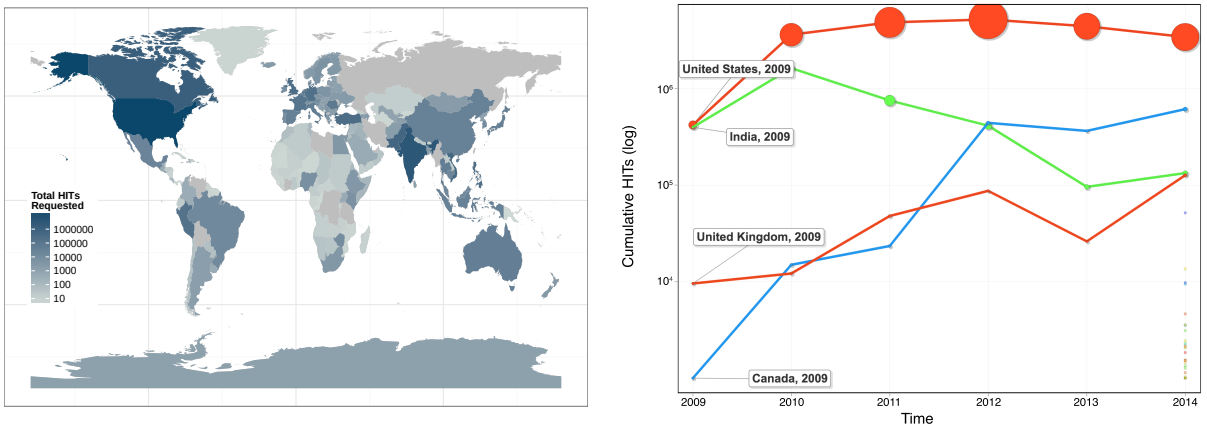


Figure 3: HITs with specific country requirements. On the left-hand side, the countries with the most HITs dedicated to them. On the right-hand side, the time evolution (x-axis) of country-specific HITs with volume (y-axis) and reward (size of data point) information.

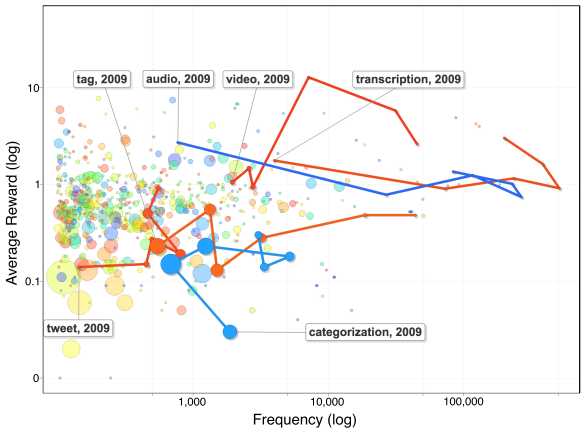


Figure 2: The use of keywords to annotate HITs. *Frequency* corresponds to how many times a keyword was used, and *AverageReward* corresponds to the average monetary reward of batches that listed the keyword. The size of the bubbles indicates the average batch size.

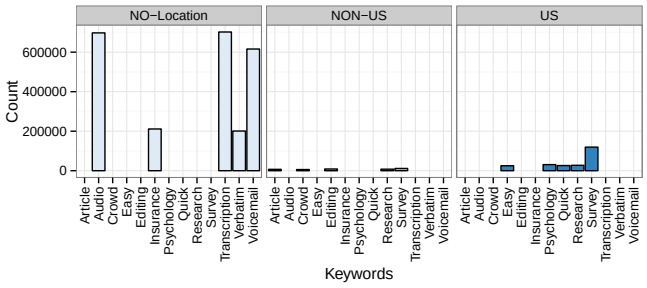


Figure 4: Keywords for HITs restricted to specific countries.

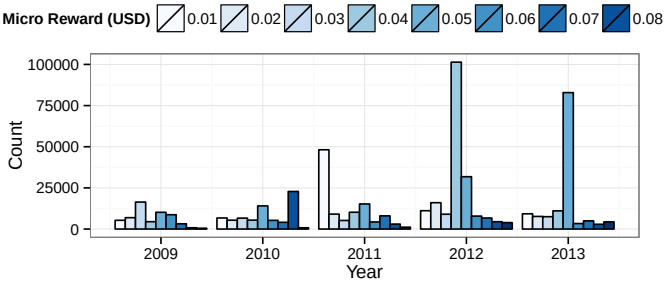


Figure 5: Popularity of HIT reward values over time.

that were posted in the considered time period did not require any specific worker location. 86% of those which did, imposed a constraint requesting US-based workers.

Figure 4 shows the top keywords attached to HITs restricted to specific locations. We observe that the most popular keywords (i.e., 'audio' and 'transcription') do not require country-specific workers. We also note that US-only HITs are most commonly tagged with 'survey'.

HIT Reward Analysis.

Figure 5 shows the most frequent rewards assigned to HITs over time.[2] We observe that while in 2011 the most popular reward was $0.01, recently HITs paid $0.05 are getting more frequent. This can be explained both by how workers search for HITs on Amazon MTurk and by the Amazon MTurk fee scheme. Requesters now prefer to publish more complex HITs possibly with multiple questions in them and grant a higher reward: This also attracts those workers who are not willing to complete a HIT for small rewards and reduces the fees paid to Amazon MTurk, which are computed based on the number of HITs published on the platform.

[2]Data for 2014 has been omitted as it was not comparable with other year values.

Requester Analysis.

In order to be sustainable, a crowdsourcing platform needs to retain requesters over time or get new requesters

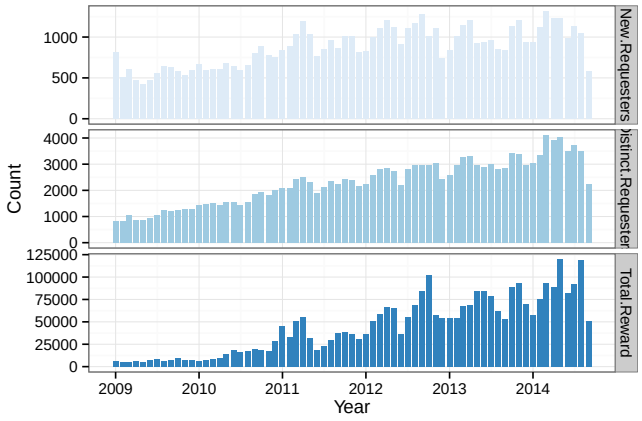


Figure 6: Requester activity and total reward on the platform over time.

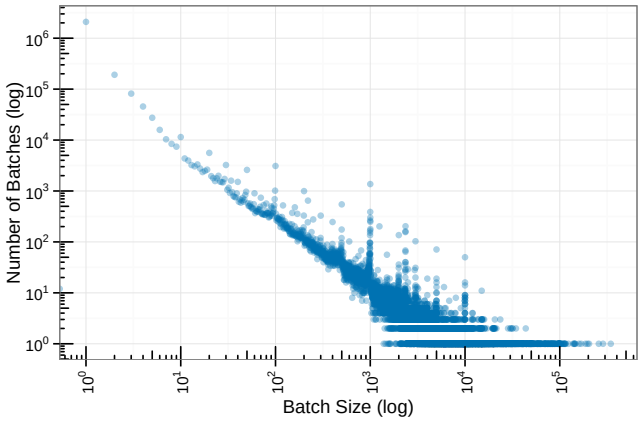


Figure 7: The distribution of batch sizes.

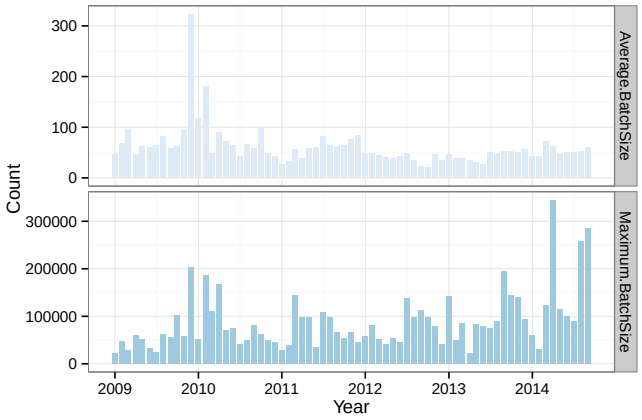


Figure 8: Average and maximum batch size per month. The monthly median is 1.

to replace those who do not publish HITs anymore. Figure 6 shows the number of new requesters who used Amazon MTurk and the overall number of active requesters at a certain point in time. We can observe an increasing number of active requesters over time and a constant number of new requesters who join the platform (at a rate of 1,000/month over the last two years).

It is also interesting to look at the overall amount of reward for HITs published on the platform, as platform revenues are computed as a function of HIT reward. From the bottom part of Figure 6, we observe a linear increase in the total reward for HITs on the platform. Interestingly, we also observe some seasonality effects over the years, with October being the month with the highest total reward and January or February being the month with minimum total reward.

HIT Batch Size Analysis.

When a lot of data needs to be crowdsourced (e.g., when many images need to be tagged), multiple tasks containing similar HITs can be published together. We define a batch of HITs as a set of similar HITs published by a requester at a certain point in time.

Figure 7 shows the distribution of batch sizes in the period from 2009 to 2014. We can observe that most of the batches were of size 1 (more than 1M), followed by a long tail of larger, but less frequent, batch sizes.

Figure 8 shows how batch size has changed over time. We observe that the average batch size has slightly decreased. The monthly median is 1 (due to the heavily skewed distribution). Another observation that can be made is that in 2014 very large batches containing more that 200,000 HITs have appeared on Amazon MTurk.

4. LARGE-SCALE HIT TYPE ANALYSIS

In this section, we present the results of a large-scale analysis of the evolution of HIT types published on the Amazon MTurk platform. For this analysis, we used the definition of HIT types proposed by [10] in which authors perform an extensive study involving 1,000 crowd workers to understand their working behavior, and categorize the types of tasks that the crowd perform into six top-level "goal-oriented" tasks, each containing further sub-classes. We briefly describe the six top-level classes introduced by [10] below.

- Information Finding (IF): Searching the Web to answer a certain information need. For example, "Find the cheapest hotel with ocean view in Monterey Bay, CA".
- Verification and Validation (VV): Verifying certain information or confirming the validity of a piece of information. Examples include checking Twitter accounts for spamming behaviors.
- Interpretation and Analysis (IA): Interpreting Web content. For example, "Categorize product pictures in a predefined set of categories", or "Classify the sentiment of a tweet".
- Content Creation (CC): Generating new content. Examples include summarizing a document or transcribing an audio recording.
- Surveys (SU): Answering a set of questions related to a certain topic (e.g., demographics or customer satisfaction).
- Content Access (CA): Accessing some Web content. Examples include watching online videos or clicking on provided links.

4.1 Supervised HIT Type Classification

Using the various definitions of HIT types given above, we trained a supervised machine learning model to classify HIT types based on their metadata. The features we used to train the Support Vector Machine (SVM) model are: HIT title, description, keywords, reward, date, allocated time, and batch size.

To train and evaluate the supervised model, we created labelled data: We uniformly sampled 5,000 HITs over the entire five-year dataset and manually labelled their type by means of crowdsourcing. In detail, we asked workers on MTurk to assign each HIT to one of the predefined classes by presenting them with the title, description, keywords, reward, date, allocated time, and batch size for the HIT. The instructions also contained the definition and examples for each task type. Workers could label tasks as 'Others' when unsure or when the HIT did not fit in any of the available options.

After assigning each labelling HIT to three different workers in the crowd, a consensus on the task type label was reached in 89% of the cases (leaving 551 cases with no clear majority). A consensus was reached when at least two out of three workers agreed on the same HIT type label. The other cases, that is, when the workers provided different labels or when they where not sure about the HIT type, have then been removed from our labelled dataset.

Using the labelled data, we trained a multi-class SVM classifier for the 6 different task types and evaluated its quality with 10-fold cross validation over the labelled dataset. Overall, the trained classifier obtained a Precision of 0.895, a Recall of 0.899, and an F-Measure of 0.895. Most of the classifier errors (i.e., 66 cases) were caused by incorrectly classifying IA instances as CC jobs.

Performing feature selection for the HIT type classification problem, we observed that the best features based on information gain are the HIT allotted time and reward: This indicates that HITs of different types are associated with different levels of reward as well as different task durations (i.e., longer and better paid tasks versus shorter and paid worse). The most distinctive keywords for identifying HIT types are 'transcribe', 'audio', and 'survey', which clearly identify CC and SU HITs.

Using the classifier trained over the entire labelled dataset, we then performed a large-scale classification of the types for all 2.5M HITs in our collection. This allows us to study the evolution of the task types over time on the Amazon MTurk platform, which we describe next.

4.2 Task Type Popularity Over Time

Using the results of the large-scale classification of HIT types, we analyze which types of HITs have been published over time. Figure 9 shows the evolution of task types published on Amazon MTurk. We can observe that, in general, the most popular type of task is Content Creation. In terms of observable trends, we note that–while there is a general increase in the volume of tasks on the platform—CA tasks have been decreasing over time. This can be explained by the enforcement of Amazon MTurk terms of service, which state that workers should not be asked to create accounts on external websites or be identified by the requester. In the last three years, SU and IA tasks have seen the biggest increase.

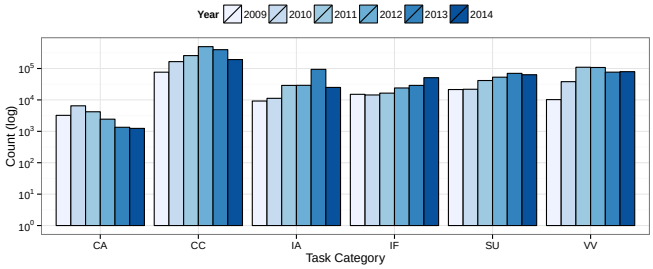


Figure 9: Popularity of HIT types over time.

5. ANALYZING THE FEATURES AFFECTING BATCH THROUGHPUT

Next, we turn our attention to analyzing the factors that influence the progress (or the pace) of a batch, how those factors influence each other and how their importance changes over time.

In order to conduct this analysis, we carry out a prediction experiment on the batch's *throughput*, that is, the number of HITs that will be completed for a given batch within the next time frame of 1 hour (i.e., the $DIFF\_HIT$ feature is the target class). Specifically, we model this task as a regression problem using 29 features; some of them were used in the previous section to classify the HIT type; we describe the remaining ones in Appendix A.

5.1 Throughput Prediction

To predict the throughput of a batch at time T, we train a Random Forest Regression model with samples taken in the range $[T - \delta, T)$ where δ is the size of the time window that we are considering directly prior to time T. The rationale behind this approach is that the throughput should be directly correlated to the current and recent market situations.

We considered data from June to October 2014, and hourly observations (see Section 3.1), from which we uniformly sampled 50 test time points for evaluation purposes. In our experiments, the best prediction results, in terms of R-squared[3], were obtained using $\delta = 4hours$. For that window, our predicted versus actual throughput values are shown in Figure 10. The figure suggests that the prediction works best for larger batches having a large momentum.

In order to understand which features contribute significantly to our prediction model, we proceed by feature ablation. For this experiment, we computed the prediction evaluation score R-squared, for 1,000 randomly sampled test time points and kept those where the prediction worked reasonably, i.e., having R-squared> 0, that is 327 samples. Next, we rerun the prediction on the same samples by removing one feature at a time. The results revealed that the features $HIT\_available$ (i.e., the number of tasks in the batch) and $Age\_minutes$ (i.e., how long ago the batch was created) were the only ones having a statistically significant impact on the prediction score with $p < 0.05$ and $p < 0.01$ respectively.

[3]`http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html`

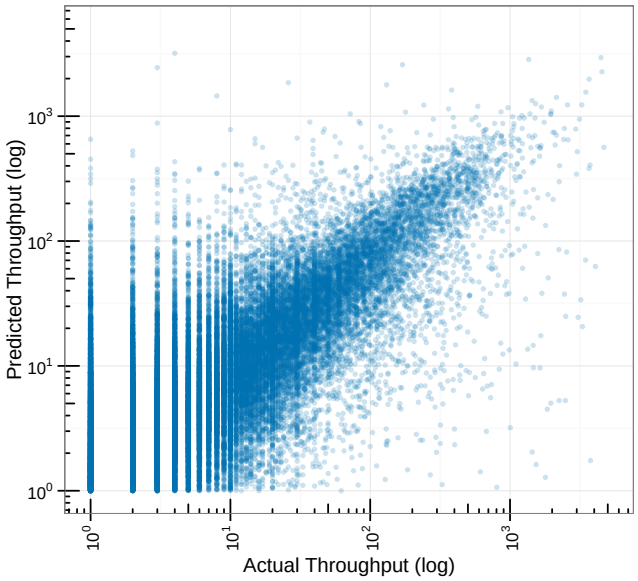


Figure 10: Predicted vs actual batch throughput values for $\delta = 4hours$. The prediction works best for larger batches having a large momentum.

Table 1: Gini importance of the top 2 features used in the prediction experiment. A large mean indicates a better overall contribution to the prediction. A positive slope indicates that the feature is gaining in importance when the considered time window is larger.

Feature	mean	stderr	slope	intercept
HIT_available	29.8606	13.4247	-0.0257	34.4940
Age_minutes	12.9087	8.1967	-0.0050	13.8181

5.2 Features Importance

In order to better grasp the characteristics of the batch throughput, we examine the computed Gini importance of the features [3]. In this experiment, we varied the training time frame δ from 1 hour to 24 hours for each tested time point. Figure 11 shows the contribution of our 2 top features (as concluded from the previous experiment, i.e., *HIT_available* and *Age_minutes*) and how their importances varied when we increased the training time-frame. These features are again listed in Table 1, the slope indicates whether the feature is gaining importance over time (positive value) or decreasing in importance (negative value).

The most important feature is *HIT_available*, that is, the current size of the batch. Indeed, as observed by previous work, larger batches tend to attract more workers [12, 9]. This feature becomes less important when we consider longer periods, partly because of noise, and because other features start to encode additional facts. On the other hand, *Age_minutes* importance suggests that the crowd is sensitive to newly posted HITs, or how *fresh* the HITs are. To better understand this phenomenon, we conduct an analysis on what attracts the workforce to the platform in the next section.

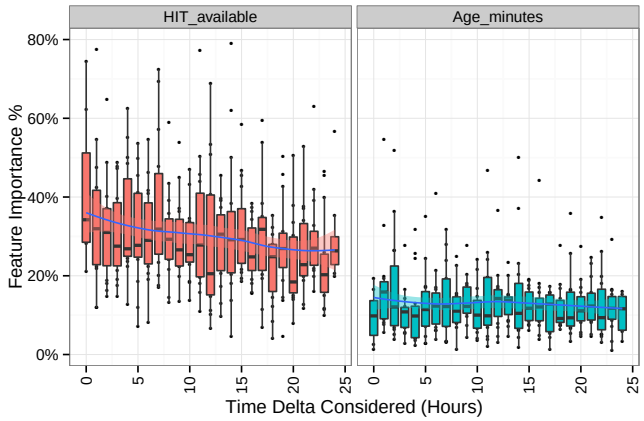


Figure 11: Computed feature importance when considering a larger training window for batch throughput prediction.

6. MARKET ANALYSIS

Finally, we study the demand and supply of the Amazon MTurk marketplace. In the following, we define *Demand* as the number of new tasks published on the platform by the requesters. In addition, we compute the average reward of the tasks that were posted. Conversely, we define *Supply* as the workforce that the crowd is providing, concretized as the number of tasks that got completed in a given time window by the workers. In this section we use hourly collected data for the time period spanning June to October 2014.

6.1 Supply Attracts New Workers

We start by analyzing how the market reacts when new tasks arrive on the platform, in order to understand the degree of elasticity of the supply. If the supply of work is inelastic, the amount of work done over time should be independent of the demand for work. So, if the amount of tasks available in the market ("demand") increases, then the percentage of work that gets completed in the market should drop, as the same amount of "work done" gets split among a higher number of tasks. To understand the elasticity of the supply, we regressed the percentage of work done in every time period (measured as the percentage of HITs that are completed) against the number of new HITs that are posted in that period. Figure 12 shows the scatterplot for those two variables.

Our data reveals that an increase in the number of arrived HITs is positively associated with a higher percentage of completed HITs. This result provides evidence that the new work that is posted is more attractive than the tasks previously available in the market, and attracts "new work supply".[4]

Our regression[5] of the "Percent Completed" against "Hits Arrived (in thousands)" indicates an intercept of 2.5 and a slope of 0.05. To put these numbers in context: On average, there are 300K HITs available in the market at any given time, and on average 10K new HITs arrive every hour. The

[4]From the data available, it is not possible to tell whether the new supply comes from distinct workers, from workers that were idle, or from an increased productivity of existing workers.

[5]We use Ordinary Least Squares regression.

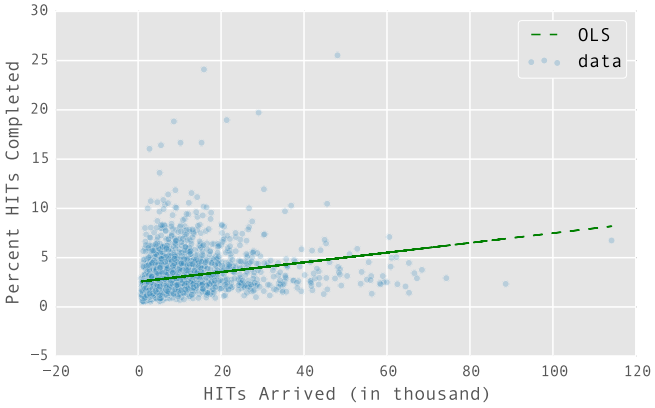


Figure 12: The effect of new arrived HITs on the work supplied. Here, the supply is expressed as the percentage of HITs completed in the market.

intercept of 2.5 means that 2.5% of these 300K HITs (i.e., 7.5K per hour) get completed, as a baseline, assuming that no new HIT gets posted. The slope is 0.05, meaning that if 10K new HITs arrive within an hour, then the completion ratio increases by 0.5%, to 3% (i.e., 9K HITs per hour). When 50K new HITs arrive within an hour, then the completion percentage increases to 5% indicating that 15K to 20K HITs get completed. In other words, approximately 20% of the *new* demand gets completed within an hour of being posted, indicating that new work has almost 10x higher attractiveness for the workers than the remaining work that is available on the platform. This result could be explained by how tasks are presented to workers by Amazon MTurk. Workers, when not searching for tasks using specific keywords, are presented with the most recently published tasks first.

6.2 Demand and Supply Periodicity

On the demand side, some requesters frequently post new batches of recurrent tasks. Hence, we are interested in the periodicity of such demand in the marketplace and the supply it drives. To look in this, we consider both the time-series of available HITs and the rewards completed.

First, we observe that the demand exhibits a strong weekly periodicity, which is reflected by the autocorrelation that we compute from the number of available HITs on Amazon Mturk (See Figure 13a and 13c). The market seems to have a significant memory that lasts for approximately 7-10 days. This indicates that future transactions are highly predictable using simple algorithms [8].

Conversely, and to check for the periodicity in the supply, we compute an autocorrelation on the weekly moving average of the completed HITs reward. Figure 13b and 13d show that there is a strong weekly periodicity effect, as we observe high values in the range 0-250 hours.

7. DISCUSSION

In this section, we summarize the main findings of our study and present a discussion of our results. We extracted several trends from the five years data, summarized as follows:

- Tasks related to audio transcription have been gaining momentum in the last years and are today the most popular tasks on Amazon MTurk.
- The popularity of Content Access HITs has decreased over time. Surveys are however becoming more popular over time especially in the US.
- While most HITs do not require country-specific workers, most of such HITs require US-based workers.
- HITs that are exclusively asking for workers based in India have strongly decreased over time.
- Surveys are the most popular type of HITs for US-based workers.
- The most frequent HIT reward value has increased over time, and reaches $0.05 in 2014.
- New requesters constantly join Amazon MTurk, making the total number of active requesters and the available reward increase over time.
- The average HIT batch size has been stable over time; however, very large batches have recently started to appear on the platform.

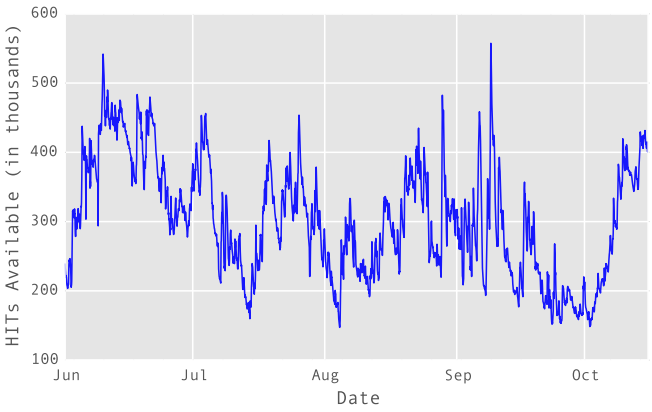
Our batch throughput prediction (Section 5) indicates that the throughput of batches can be best predicted based on the number of HITs available in the batch,i.e., its size; and its freshness, i.e., for how long the batch has been on the platform.

Finally, we analyzed Amazon MTurk as a marketplace in terms of demand (new HITs arriving) and supply (HITs completed). We observed some strong weekly periodicity both in demand and in supply. We can hypothesize that many requesters might have repetitive business needs following weekly trends, while many workers work on Amazon MTurk on a regular basis during the week.

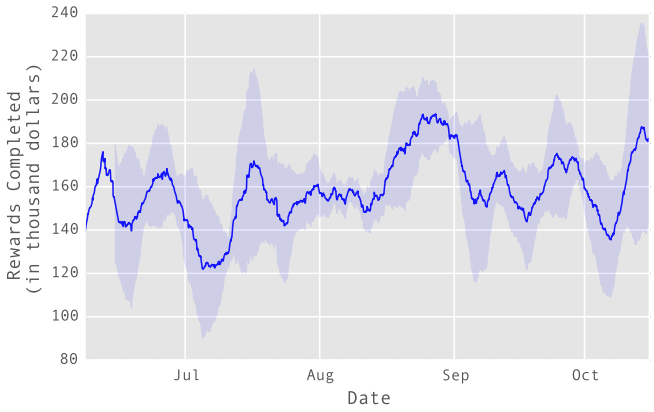
8. CONCLUSIONS

We studied data collected from a popular micro-task crowdsourcing platform, Amazon MTurk, and analyzed a number of key dimensions of the platform, including: topic, task type, reward evolution, platform throughput, and supply and demand. The results of our analysis can serve as a starting point for improving existing crowdsourcing platforms and for optimizing the overall efficiency and effectiveness of human computation systems. The evidence presented above indicate how requesters should use crowdsourcing platforms to obtain the best out of them: By engaging with workers and publishing large volumes of HITs at specific points in time.

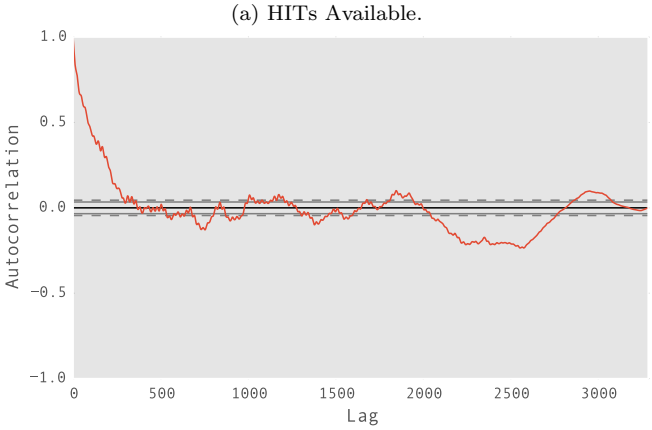
Future research based on this work might look at different directions. On one hand, novel micro-task crowdsourcing platforms need to be designed based on the findings identified in this work, such as the need for supporting specific task types like audio transcription or surveys. Additionally, analyses that look at specific data could provide a deeper understanding of the micro-task crowdsourcing universe. Examples include per-requester or per-task analyses of the publishing behavior rather than looking at the entire market evolution as we did in this work. Similarly, a worker-centered analysis could provide additional evidence of the existence of different classes of workers, e.g., full-time vs casual workers, or workers specializing on specific task types as compared to generalists who are willing to complete any available task. While a requester-centered analysis would consider information about the requesters' reputation, pricing and HIT types.



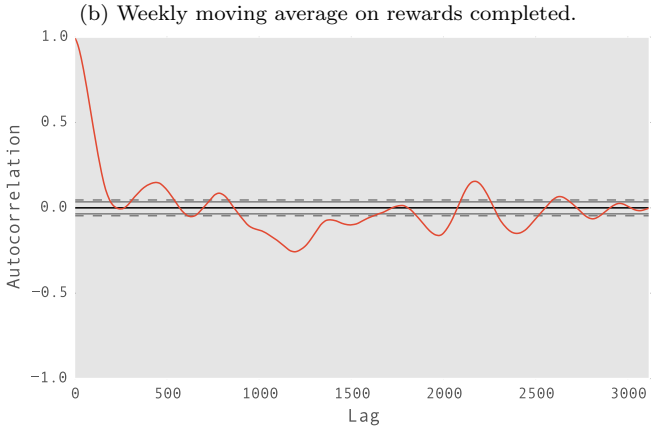
(a) HITs Available.



(b) Weekly moving average on rewards completed.



(c) Autocorrelation on HITs available.



(d) Autocorrelation on moving average of rewards completed.

Figure 13: Computed autocorrelation on the number of HITs available and on the weekly moving average of the completed reward (N.B., autocorrelation's Lag is computed in Hours). In both cases, we clearly see a weekly periodicity (0-250 Hours).

9. ACKNOWLEDGMENTS

We thank the reviewers for their helpful comments. This work was supported by the Swiss National Science Foundation under grant number PP00P2_128459.

APPENDIX

A. MACHINE LEARNING FEATURES USED TO PREDICT THROUGHPUT

The following is the list of features associated to each batch. We used these features in our machine learning approach to predict batch throughput for the next hourly observation (see Section 5):

- **HIT_available:** Number of available HITs in the batch.
- **start_time:** The time of an observation.
- **reward:** HIT Reward in USD.
- **description:** String length of the batch's description.
- **title:** String length of the batch's title.
- **keywords:** Keywords (space separated).
- **requester_id:** ID of the requester.
- **time_alloted:** Time allotted per task.
- **tasktype:** Task class (as per our classification in 4).
- **ageminutes:** Age since the Batch was posted (minutes).
- **leftminutes:** Time left before expiration (minutes).
- **location:** The requested worker's Location (e.g., US).
- **totalapproved:** Batch requirement on the number of total approved HITs.
- **approvalrate:** Batch requirement on the percentage of workers approval.
- **master:** Worker is a master.
- **hitGroupsAvailableUI:** Number of batches as reported on Mturk dashboard.
- **hitsAvailableUI:** Number of HITs available as reported on Mturk dashboard.
- **hitsArrived:** Number of new HITs arrived.
- **hitsCompleted:** Number of HITs completed.
- **rewardsArrived:** Sum of rewards associated with the HITs arrived.
- **rewardsCompleted:** Sum of rewards associated with the HITs completed.
- **percHitsCompleted:** Ratio of HITs completed and total HITs available.
- **percHitsPosted:** Ratio of new HITs arrived and total HITs available.
- **diffHits:** hitsCompleted-hitsArrived.
- **diffHitsUI:** Difference in HITs observed from Mturk dashboard.
- **diffGroups:** Computed difference in number of completed and arrived batches.
- **diffGroupsUI:** Difference in number of completed and arrived batches observed from Mturk dashboard.

- **diffRewards:** Difference in rewards = (rewardsArrived-rewardsCompleted).
- **DIFF_HIT:** Number of HITs completed since the last observation.

B. REFERENCES

[1] O. Alonso and S. Mizzaro. Using crowdsourcing for TREC relevance assessment. *Inf. Process. Manage.*, 48(6):1053–1066, 2012.

[2] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci. Choosing the right crowd: Expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 637–648, New York, NY, USA, 2013. ACM.

[3] L. Breiman and A. Cutler. Random Forests. `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`. Accessed: 2015-03-04.

[4] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. 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, pages 469–478, New York, NY, USA, 2012. ACM.

[5] D. E. Difallah, M. Catasta, G. Demartini, and P. Cudré-Mauroux. Scaling-up the Crowd: Micro-Task Pricing Schemes for Worker Retention and Latency Improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[6] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux. Pick-a-crowd: Tell me what you like, and i'll tell you what to do. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 367–374, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[7] S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. *Human Computation*, 11, 2011.

[8] J. D. Farmer, A. Gerig, F. Lillo, and S. Mike. Market Efficiency and the Long-Memory of Supply and Demand: Is Price Impact Variable and Permanent or Fixed and Temporary. *Quant. Finance*, 6(2):107–112, 2006.

[9] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering Queries with Crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 61–72, New York, NY, USA, 2011. ACM.

[10] U. Gadiraju, R. Kawase, and S. Dietze. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, pages 218–223, New York, NY, USA, 2014. ACM.

[11] Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. *PVLDB*, 7(14):1965–1976, 2014.

[12] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21, Dec. 2010.

[13] L. C. Irani and M. S. Silberman. Turkopticon: Interrupting Worker Invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 611–620, New York, NY, USA, 2013. ACM.

[14] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 1301–1318, New York, NY, USA, 2013.

[15] P. Kucherbaev, S. Tranquillini, F. Daniel, F. Casati, M. Marchese, M. Brambilla, and P. Fraternali. Business processes for the crowd computer. In M. La Rosa and P. Soffer, editors, *Business Process Management Workshops*, volume 132 of *Lecture Notes in Business Information Processing*, pages 256–267. Springer Berlin Heidelberg, 2013.

[16] A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1003–1012, New York, NY, USA, 2012. ACM.

[17] J. Mortensen, M. A. Musen, and N. F. Noy. Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA*, 2013.

[18] A. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom. Human-assisted graph search: It's okay to ask questions. *Proc. VLDB Endow.*, 4(5):267–278, Feb. 2011.

[19] C. Sarasua, E. Simperl, and N. F. Noy. Crowdmap: Crowdsourcing ontology alignment with microtasks. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part I*, ISWC'12, pages 525–541, Berlin, Heidelberg, 2012. Springer-Verlag.

[20] M. S. Silberman, L. Irani, and J. Ross. Ethics and tactics of professional crowdwork. *XRDS*, 17(2):39–43, Dec. 2010.

[21] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, Aug. 2008.

[22] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 55–64, New York, NY, USA, 2006. ACM.

[23] M. Vukovic. Crowdsourcing for enterprises. In *Services-I, 2009 World Conference on*, pages 686–692. IEEE, 2009.

[24] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, July 2012.

[25] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 229–240, New York, NY, USA, 2013. ACM.