mClerk: Enabling Mobile Crowdsourcing in Developing Regions

Aakar Gupta[1], William Thies[2], Edward Cutrell[2] and Ravin Balakrishnan[1]
[1] University of Toronto, Canada
{aakar, ravin}@dgp.toronto.edu
[2] Microsoft Research India
{thies, cutrell}@microsoft.com

ABSTRACT
Global crowdsourcing platforms could offer new employment opportunities to low-income workers in developing countries. However, the impact to date has been limited because poor communities usually lack access to computers and the Internet.

This paper presents mClerk, a new platform for mobile crowdsourcing in developing regions. mClerk sends and receives tasks via SMS, making it accessible to anyone with a low-end mobile phone. However, mClerk is not limited to text: it leverages a little-known protocol to send small images via ordinary SMS, enabling novel distribution of graphical tasks. Via a 5-week deployment in semi-urban India, we demonstrate that mClerk is effective for digitizing local-language documents. Usage of mClerk spread virally from 10 users to 239 users, who digitized over 25,000 words during the study. We discuss the social ecosystem surrounding this usage, and evaluate the potential of mobile crowdsourcing to both deliver and derive value from users in developing regions.

Author Keywords
ICTD; Mobile Crowdsourcing; Microtasks; Digitization

ACM Classification Keywords
H.5.2. [**User Interfaces**]; K.4.2 [**Social Issues**]: Employment

General Terms
Design, Human Factors

INTRODUCTION
Paid crowdsourcing platforms (such as Amazon Mechanical Turk) are a potential means to improve the livelihoods of low-income workers in developing countries. By eliminating the need for formal contracts and co-location between employer and employee, paid crowdsourcing could lower the barrier-to-entry in the global marketplace and provide a higher rate of pay than is available locally. In addition, crowdsourced tasks can be completed on a flexible schedule, offering workers the opportunity to earn supplemental revenue during their commute to work or during other idle moments of the day.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

However, to date paid crowdsourcing has not delivered on its potential to impact poor communities in developing regions. While about one third of the workers on Mechanical Turk are based in India [20], they are usually college-educated and have an income that is more than double the Indian average [17]. Prior work has identified two key barriers that prevent lower-income individuals from earning money via crowdsourcing [17, 22]. First, they lack access to technology – computers, Internet, and smart phones – that are typically needed to complete paid tasks. While public centers such as Internet cafes may be available, the price of access is often comparable to the wages earned online [17]. Though services such as txteagle [1] have sought to distribute textual tasks to low-end mobile phones, text remains limiting; many real-world tasks require access to images. The second barrier to the uptake of paid crowdsourcing in low-income groups is workers' education: they lack the skills to compose English sentences, or to understand complex English instructions. Very few tasks leverage their unique skills, for example, their knowledge of local languages and customs.

This paper presents the design and evaluation of mClerk, a new platform for paid crowdsourcing amongst low-income workers in developing countries. Our system leverages two insights to overcome the limitations of prior platforms. First, we enable image-based tasks to be distributed to low-end mobile phones, using a little-known protocol that sends small bitmapped images via ordinary SMS messages. Usage of this protocol enables low-income workers to participate in the system using devices that they already own. Responses are also sent via SMS, which is very affordable. Second, we identify an important and large-scale problem – digitization of local-language text – that is uniquely suited to the skills of low-income workers and the capabilities of our platform. To complete this task, users receive an image of a word via SMS, and send back the typed version. To handle local-language fonts, users transliterate the word in English, and our system later converts it to the local font.

We instantiate these ideas in the form of a crowdsourced platform for digitizing local-language documents. In addition to the components described above, our system automatically segments scanned forms into individual words, checks correctness by duplicating tasks across multiple workers, and pays workers via mobile airtime credits. Altogether, our platform offers novel benefits to both requesters (first work that addresses crowdsourced

digitization of local-language text) as well as workers (first paid data-entry opportunity from low-end phones).

To demonstrate the viability of our platform, we conducted a 5-week, real-world deployment in and around Bangalore, India. We observed a viral uptake of the system: after recruiting 10 users ourselves, the usage spread via referrals to 239 users, who digitized over 25,000 words in the course of the study. We document a vibrant social ecosystem that emerged amongst the users of the system, spanning students, shopkeepers, and laborers. The system is also attractive from the standpoint of requesters: it digitized words with an accuracy of 90.1% and a cost of INR 0.2 to 0.5 (USD 0.004 to 0.01)[1] per word, which is comparable to the leading market alternatives.

These findings indicate that there is a large untapped potential for paid crowdsourcing to simultaneously benefit and benefit from low-income workers in developing regions. Moreover, our results were enabled by a novel interaction technique: the transmission of pictures via low-cost SMS messages.

RELATED WORK

Microtasking in Developing Regions

The most visible platform for mobile crowdsourcing in developing regions is txteagle, which has been deployed in Africa [15] and is establishing a broader user base worldwide [1]. While txteagle also operates via text messages, it has not utilized picture SMS to send graphical tasks. MobileWorks [2] is the closest counterpart to our system in terms of the user experience: users transcribe images of text on mobile phones. However, it requires the use of a mobile web browser enabled by a data connection and has so far placed its focus on the English language.

Samasource [3] also enables microtasking for marginalized workers, but instead of crowdsourcing the work, it works with local partner organizations that maintain a dedicated employee workforce supported by an Internet-enabled computer. It briefly offered an iPhone application, Give Work, to enable rich users to volunteer and verify the accuracy of tasks completed by Samasource workers. Ushahidi [4] enables mobile crowdsourcing of crisis information. In addition, there are many other web crowdsourcing services that likely draw some users from developing regions [16], but require computer access. Some of them are CrowdFlower, CloudCrowd, Smartsheet, CrowdSifter, LeadVine, LiveWork and LogoTournament.

mClerk is distinguished from these prior efforts in two respects. It is the first system to utilize picture SMS for low-cost distribution of graphical tasks, and, as far as we know, it is the first system to demonstrate large-scale digitization of local-language texts (which lack font support on common devices) via crowdsourcing.

[1] Throughout this paper, we use an exchange rate of 45 Indian Rupees (INR) to one US dollar (USD).

Digitization

Captricity [5] is a service that promises digitization of paper forms by segmenting them and using OCR to generate approximations, followed by crowdsourcing verification tasks on MTurk or in-sourcing them to the organization's own workers. However, it does not cater to local language documents and does not have a mobile component. reCAPTCHA digitizes scanned English documents as a beneficial by-product of a human verification task [11].

THE MCLERK SYSTEM

In essence, mClerk starts with a scan of a paper document, segments it into word images, sends each image via SMS to users' phones, receives back the users' responses, probabilistically verifies them, pays the users and aggregates responses into a digital document. It has four modules: image segmentation software, a mobile crowdsourcing platform, word aggregation code, and a payment mechanism. The software was implemented in C#. Before detailing each of the modules, we explain the mobile phone protocols that enable us to send images via SMS.

Images via SMS

Nokia's Smart Messaging (SM) [9] and Ericsson's EMS [8] are device dependent protocols that predate MMS and support sending binary picture messages via SMS, albeit with restrictions. The pictures are restricted to 74x28 pixels for SM and 64x16 pixels for EMS. In our study, the usage was dominated by Nokia users (SM), consequently we will focus on SM. The SM picture message is actually three concatenated binary SMS messages (Fig. 1). As the green text depicts in Fig. 1(c), we can also send text with the image in the same message. The message picture here displays white text on black background, even when our original processed image is black on white. This varies from device to device.

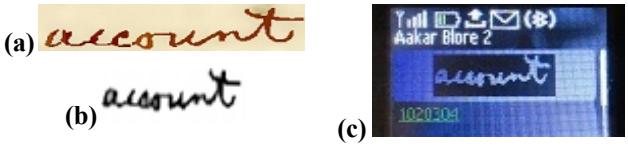

Figure 1. (a) A word in the original image (b) The word in SM format (c) A sample SM message.

Image Segmentation

Our current software, partially based on an algorithm by Arivazaghan et al. [13], places its focus on paper documents that contain textual data. It also works with general documents which have user-defined sections for textual data, which can be extracted once their locations are marked. The system segments scanned pages of handwritten or printed text into separate words, considering various issues such as multiple ink and background shades, paper skew, text skew and salt and pepper noise. The segmented word images are then binarized and resized.

Mobile Crowdsourcing

The segmented word images are sent to the users' mobile phones, where users are expected to key in each word and

send back the text. A major problem is that the majority of phones lack local language font support. Although Hindi fonts are ubiquitous in mobiles phones, India has 22 official languages and the local language in our study, Kannada, is not uniformly supported on phones. Moreover, even for the Hindi script, we observed that users found typing to be tedious and error-prone, especially for intricate vowel marks. To overcome these hurdles, we propose a solution where users key in and send the best English transliteration of the word that they can. The user needs to have a basic functional knowledge of English. Although this limits accessibility to some extent, familiarity with "broken English" is fairly common in India and this is not an overbearing hurdle in our context.

The verification of responses in microtasking systems is typically done by comparing multiple responses for the same task – for instance, reCAPTCHA admits a response only if there is an equivalent response from a 2nd user for the same word. We follow the same protocol where each word is sent to two users. However, in our case, since the users are sending back transliterations, there could be a large number of cases where similar but distinct transliterations point to the same word. For instance, "Nammaa" and "Namma" are the same word in Kannada. To handle this problem, we modified our algorithm so that two responses are considered equivalent if both of them transliterate back to the same word in the local language. We make use of Google's transliteration API [6] to convert from English to Kannada characters. After this, verification follows the usual procedure where two equivalent responses for the same word are considered to be correct. If the first two responses to the same word do not match, then we send the word to a third person and continue this process until we get two equivalent responses. Finally, all the responses in agreement are aggregated into a digital document.

Payment Mechanism

Compensation is in the form of mobile airtime as it can be administered remotely and does not require a bank account. Most of the mobile network operators in India have capped the minimum recharge/top-up amount at INR 10 (USD 0.22). Hence, we make the airtime payments in chunks of INR 10, after the user has successfully completed tasks worth that amount (we discuss payment rates later). The users get paid for correct responses only.

Because there are no services that allow automated payments for all network operators (there are twelve!) in India, we opted for a manual system of payments. In India, there are ubiquitous "recharge shops" that administer airtime payments in both rural and urban settings. We partnered with one such shop owner. He was emailed a daily list of payments, which he administered manually on the same day in exchange for a small commission.

ITERATIVE PROTOTYPING

To send the messages to the users there were 2 possible protocols: 1) to send a fixed number of messages in a group to each user daily, or 2) to send messages one by one, i.e., the next message is not sent until the prior response is received. When messages go in a group we cannot expect a user to reply in the same sequence as received, without missing one. Therefore, the user has to type in a unique message id first, followed by space and the word. The second protocol does not require an identifier, but it runs the risk of halting a user's progress even if a single picture message or its response is delayed or lost.

To determine which protocol is best, we conducted a pilot study where 2 groups of 3 users each were assigned a given protocol. We found that the responses from those receiving grouped messages were erratic. Some replies had the word typed in first, followed by the id in a new line or hyphen instead of space. Despite clear instructions, the inevitable errors sealed the argument in favor of one-by-one message sending, as the error probability is low. A big economic question and a potential limitation was the cost of message sending for the user, which ranged from INR 0.0 to 0.5 per SMS. During the pilot we surprisingly found that all our users had *free* SMS schemes which allowed them to send up to 500 messages free daily and as a result they did not require any reimbursements. In the words of one user:

> "We exchange jokes, shayari (poetry) daily. Using free SMS, I forward them to all my friends. It is a good deal."

USER STUDY

Besides questions about performance and accuracy, the most pertinent question addressed by our study is whether people will adopt mClerk and use it willingly, of their own volition, in a real-world setting. Can mClerk mobilize a critical mass of active users that is required for such a system to function? Further, if the answer to this question is yes, then can it be an economically self-sustainable system of value to both requesters and users, which can compete with the existing market?

We conducted a 5-week study, divided into 2 phases, with Phase 1 addressing the initial questions and Phase 2 probing the financial question. Most users were located in semi-urban India, about 4 hours outside of Bangalore. It was a good context for our study, given that people were well-versed in the local language (Kannada) and most had some knowledge of basic English. In Phase 1 (lasting 3 weeks), we paid workers INR 0.5 per correct response. Requesters would be charged INR 1.1 per word, which covers the verification and referral cost. In Phase 2 (lasting 2 weeks), we cut the payment to workers to INR 0.2 per correct response, implying a rate of INR 0.44 for requesters. As detailed later, other services for local-language digitization charge between INR 0.5 and INR 1.25 per word, so our system was competitively priced during both study phases. Our costs in sending picture messages were very nominal at INR 0.01 per SMS.

Users and Referrals

The adoption of a technological innovation is dependent on its diffusion among the members of society [19]. We

designed our intervention around this principle. Starting with a small group of 10 initial users (*core users*), we studied the spread of usage across the community. Users could refer their friends to join and they would be rewarded with 10% of the total earning of their referrals. We opted for a reward tied to the referred user's performance to boost individual throughput besides the total user count. Apart from the 10 core users, all users were enrolled via referrals.

We believe that mobile crowdsourcing will be most beneficial for low-income workers with jobs that allow them a lot of free time, such as drivers and security guards. With this in mind, we decided to have a diversified core group so as to study the diffusion in different communities. Eight users, including drivers, security guards and housekeeping staff were recruited from within a corporate office facility in urban Bangalore. These users had an average monthly income of INR 8,000 (USD 178). Upon asking these users about their friends in semi-urban locations, we obtained contact information for two additional people and registered them. Both were from a semi-urban area, 4 hours away from Bangalore. One of these users was a college student with a monthly household income of INR 9,000 (USD 200) per month, and the other was a shopkeeper earning INR 6,000 (USD 133) per month.

To refer someone, a user could simply give a missed call (calling a number and hanging up before the mobile's owner can pick up the call) on the same number that they received the tasks from. The researcher would then call back and register the new user. The essential details for registration included the referrer number, the mobile operator (to administer payments) and the phone company (to know SM/EMS). We considered using SMS based registration, but typing in all the details could potentially result in errors (as earlier with ids), resulting in the loss of a prospective user. We further extended the usage of missed calls to include troubleshooting so that if users had any queries, they could simply give a missed call and would receive a call back to resolve the issue.

Providing Feedback and Motivational Messages to Users

As the airtime was paid in chunks of INR 10, the users in Phase 1 had to complete 20 correct messages before they would receive any compensation. Previous work [14] argues for the importance of feedback in helping workers persist with the system. Since each word had to be sent to at least 2 people before getting a verified result, we could not give synchronous feedback on whether the responses were correct or not. Therefore, to motivate the users through the span of 20 correct messages, after every 10, we sent them a message like this in Kannada: *"Great going Manju! You've completed 10 more correct words. You're 10 more away from your recharge! Come On!"* The users were paid once the total amount they had earned from messaging and from referrals reached INR 10. At this point, they were sent this: *"Congratulations Manju!! You'll receive a recharge of INR 10! From your messages you earned – 9. From your friends – 1. Keep messaging!"* Additionally, if a user did not reply for 24 hours, a reminder message was sent describing their progress and urging them on to start replying again.

We noticed in our pilot study that people liked to do messaging with friends, associating a certain game element with the system. Consequently, we sent a leaderboard message at the end of the day listing the names and earnings of the day's top 5 leaders. We sent the leaderboard to a random selection of half of our users, in an attempt to measure its impact on participation and incentives. However, as users were in frequent contact with one another, almost everyone came to learn of the leaderboards.

Text Corpus and Data Collection

Initially, the paper documents were sourced from a local school, mostly children's notebooks with handwritten Kannada. This was done partly out of convenience and partly to get content that is representative of real-world data. The documents were clean enough for our software to perform good segmentation. However, due to an unexpected surge of usage (as we describe later), we ran out of this content early in our study and had to switch to a standard Kannada handwriting dataset [12]. Throughout the duration of the study we conducted periodic semi-structured interviews with users. We also gathered demographic information, such as occupation, income, mobile scheme and computer literacy, during enrollment.

RESULTS: USER RESPONSE

Phase 1

The response to the system was unanticipated, both in terms of individual productivity and social diffusion. Fig. 2 shows the user diffusion network for Phase 1, while Fig. 3 shows the total number of responses per day. In Phase 1, there were a total 221 users (85% male). They sent over 54,000 replies, digitized a total of 21,132 words, and received a total of INR 23,140 ($514.2) in payment. We received over 1,500 missed calls due to referrals and support requests. The highest number of total responses from an individual in Phase 1 was by a student, who digitized 1,717 words and is depicted as the biggest node in Fig 2. This quote from him sums up the users' enthusiasm:

> "This service is great sir! You don't need to pay anything for service activation and you get currency for sending SMS. I anyways sent 50 messages to friends daily before this. I do messages all the time, between classes with friends, in bus, even with one hand while having dinner."

The diffusion was remarkable in its spread across diverse communities, almost disregarding the contextual socio-economic barriers. Besides students, who formed 55% of our user base, there was a diverse range of user professions including shopkeepers, housewives, office workers, clerks and blue collar workers. The users' individual incomes (amongst non-students) ranged from INR 2,300 ($51) to INR 30,000 ($666) monthly (mean = 8,772 ($195), median = 8,000 ($178), SD = 5,063 ($113)). Fig. 2 shows that the spread was primarily rooted in two of the core users. Remarkably, both of them were the ones recruited from the

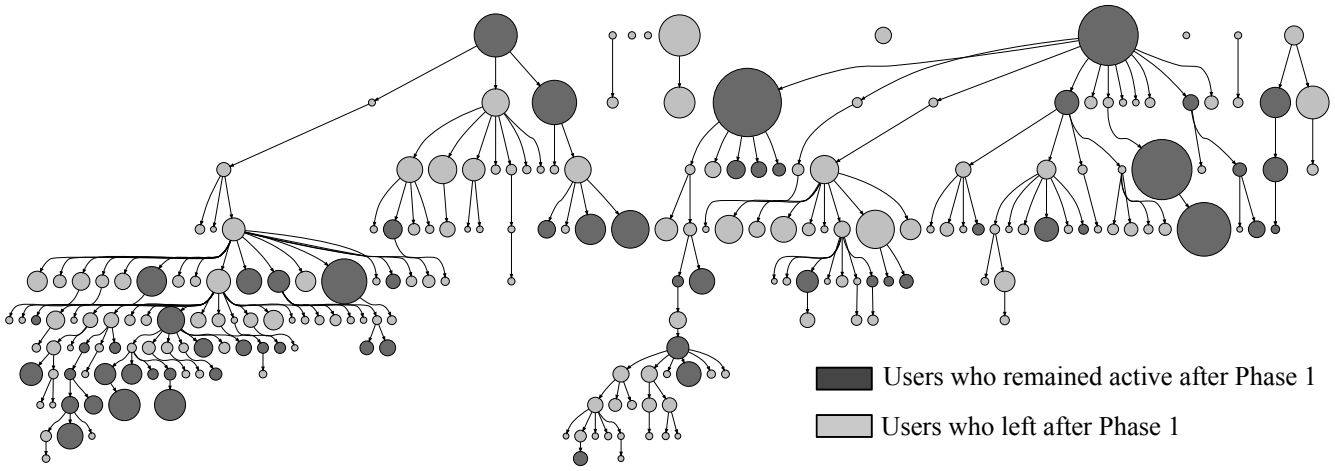

Figure 2. Users' diffusion network after Phase 1. Each node denotes a user. Each edge denotes a referral. The diameter of a node depicts the total number of responses of the user, with the minimum diameter indicating 0 responses and the maximum indicating 2221 responses.

semi-urban region, causing more than 90% of our users to be from this area.

The number of referrals on average was close to 10 per day and to an extent, depended on the number of missed calls the researchers were able to return. As usage increased, so did the missed calls – ranging from general queries about the system, slow messages during day (due to high SMS traffic) to airtime related issues and referrals. Almost all the users already had free SMS schemes; however, 4 users reported getting the schemes for this purpose alone. Activating an SMS scheme costs INR 25 to 50 (USD 0.55 to 1.11) monthly. As one user puts it:

> "I was thinking to get free SMS scheme. Then I heard about your service. I thought it is good time to get free SMS."

In contrast to free SMSs, 83% of the users reported not having a data connection (mobile Internet) on their phones. 91% did not have a computer in their home, although 78% had used a computer for basic operations. Overall, we can say that the system was effective in extending the accessibility of part-time microtasking to users who did not have access to computers or mobile Internet.

Time Pass

Most of the users reported using mClerk as a good time pass that allowed them to get mobile balance. ("Time Pass" is an Indian jargon which refers to an activity for killing time.) There were several use cases where users incorporated the system seamlessly in their routines:

> "I have to wait 20 mins for bus. I stand and do at bus-stop. I have stopped going to the recharge shop now, I get enough."

However, at times the usage effects were on the flip side:

> "We sit at back bench in class and message during lecture."

> "Earlier we [friends] used to message good-morning, good-night, jokes etc. Now no one does that. Everyone is busy."

Too Good to be True

Some users showed signs of skepticism early on, stemming from a notion that the service was "too good to be true". It was difficult for them to comprehend message sending as a valid form of paid work:

> "Is it legal? What's your profit? I don't want any trouble."

> "It is like some code sending. What do you do using this?"

In addition, the peculiarity of the task itself and the association of mobile phones with recent terrorist incidents compounded the perception of an illegal activity. The motivation behind the system was simplistically explained to the core users so that they could pass the information on. The users who understood were most efficient in allaying their friends' doubts. Some devised new, easier reasons that would convince their friends, but were often misleading:

> "System works in Nokia only, so I told my friends this system is by Nokia company to increase sales."

Ironically, CAPTCHA solving companies in India are employing thousands of workers who are probably unaware of its illegal aspects [7]. We need to watch out for such subversive uses of a system such as ours.

Social Effects

In the 1st week, there was a very strong correlation between the total number of responses and the total number of referrals for each user ($r(86) = 0.96$, $p < 0.0001$). This trend has also been observed in prior work, which concluded that lead users are active members in the social community as well [21]. The identification of lead users is potentially useful for the system. For example, as mentioned later, we used lead users to disseminate announcements (such as number change information) to others. By the end of the 2nd week, however, this correlation had become moderate ($r(191) = 0.44$, $p < 0.0001$). This trend points towards new adopters who are messaging consistently but not referring others. One reason could be the increase in the number of users whose first degree social network is already saturated. Another reason could be related to the theory [19] that early adopters are the most enthusiastic and those who follow are generally less socially forward. Interestingly, we also

observed rare cases where an earlier lead user stopped messaging, but still urged new users to join:

> "I have 10 people under me and I get enough every day. So I stopped doing messages."

We observed a moderate correlation between the total number of responses of a user and the sum of the total number of responses of his/her referrals and referrer ($r(191) = 0.49$, $p < 0.0001$). (The analysis was done after 2 weeks of usage when the number of users was 193, as the third week was non-uniform and inconsistent because of a service interruption.) This means that an individual's usage is correlated with the usage of people who are one degree away in the referral graph. Simply put, the greater the usage among friends, the greater the participation of the user. Also, users with higher usage may be more likely to invite friends in the first place.

While our initial contention was that the system would be useful for low-income workers who have a lot of free time on their hands, such as guards and drivers, the reality is that such jobs also require the person to be alone for long periods of time. In contrast, the schedules of shopkeepers (in a marketplace) and students allow them to have intermittent social interactions which evidently play a large role in their usage. Thus a better characterization of ideal users would be low-income workers who have a lot of free time *and* have professions that allow them to have social interactions. Our qualitative data contain several anecdotes which further underline the effect of social dynamics:

> "While coming back from college, all of us do messaging in the bus together and ask each other meanings of the words for fun. One time no one knew so we thought we'll ask the Kannada lecturer in college and if he does not know that will be fun.. but he knows the word."

However, there were only few such instances where this sort of active collaboration was reported. Most of the cases pointed towards an element of friendly competition:

> "Both of us did 50 [messages], he got 2 top-ups, I only got 1 … I think I did correctly … I will send more."

> "I gave my phone to my wife. She is free at home. She can do more SMS. I take it in evening when I get free with friends."

Interestingly, quite a few users gave their phones to their wives temporarily. There were other cases where a user registered his wife's phone with his name mostly because it was convenient for him after his friends told him about the system. The leaderboard was also instrumental in triggering competition. As expected, almost everyone came to know about it through their friends. Some users who were not getting the leaderboards thought of it as something they will receive upon becoming a leader and accordingly started doing more tasks. Users wanted to see their names as leaders mostly so that they could show it to their friends and did not hesitate in making extra efforts:

> "All my friends have become leaders [at least once]. Now I sleep at 12, so that I can do fast messages at night."

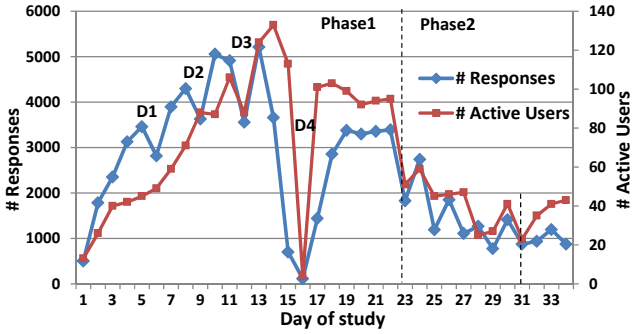

Figure 3. Total number of responses per day and number of active users vs. day of study. The dips (D1-D4) are primarily due to service outages, as detailed in the text.

In 3 weeks (barring the service interruption), there were 45 distinct users who appeared on the leaderboard at least once. This indicates that on an average, every day there were 2 users who had never been on the leaderboard before.

The reminders also proved to be remarkably effective, with 24.3% of the users responding within 1 hour of the reminder being sent to them. (This figure does not account for reminders sent after 10 PM.)

Service Outages

Fig. 3 illustrates the number of responses and number of active users (sending at least one response on a given day) over time. As a general trend, usage in Phase 1 increases as the number of users goes up day by day. However, some significant dips can be seen. The dip D1 falls on a Monday, and primarily reflects inflated usage over the weekend. D2 was caused due to overheating of our phone modem battery that rendered it useless for 2 hours. The 3rd dip was quite curious when it happened. It was Sunday and the usage was expected to grow, however some users had entirely stopped messaging. On inquiring, we found that the 1st Sunday of August is Friendship Day, which is declared as a *Black-Out Day* by some mobile operators, meaning all the free SMS schemes are deactivated for the day. This shows that users are very sensitive to the price of SMS, as expected.

The 4th and the biggest dip continued for 3 days before finally diminishing. On the 14th day, the mobile operator that we were using to send picture messages (Airtel) suddenly started dropping a large fraction of messages. After investigating for 2 days, we finally settled upon a new operator's SIM card. However, it required a few days for users to switch to the new number. We instructed users to switch via an SMS broadcast, as well as via personal contact with the lead users.

Phase 2

In Phase 2, we reduced workers' compensation to INR 0.2 per word, to probe the elasticity of the workers' supply. Including costs of verification and referrals, requesters would have to pay only INR 0.44 per word, which is lower than the price quoted by any competing service we could find in Bangalore (details given later). To compensate for the decreased payments in this phase, we also introduced

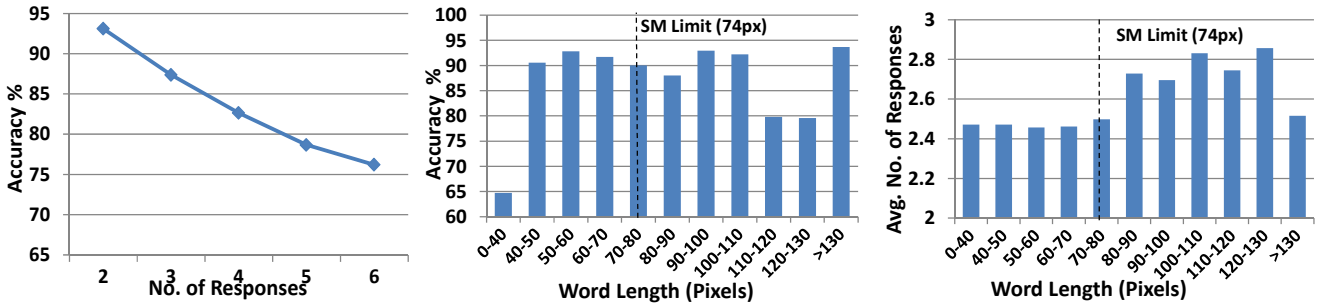

Figure 4: (a) Accuracy vs. number of responses (b) Accuracy vs. word length (c) Number of responses vs. word length

new bonuses to help motivate workers. A bonus prize of INR 30 (USD 0.60) was announced for the daily leaders. Also, 5 random jackpots – four worth INR 30 (USD 0.60) and one worth INR 50 (USD 1.11) – were added. A user had a probability of winning the jackpot on every message that they sent. While the leader bonus was to motivate the power users to do even more messages, the jackpot bonus was to motivate all users to use the system. We let 2 days pass after introducing the bonuses so that users could get habituated. Thereafter, we reduced the payments and informed the users. This phase continued for a week.

Following the reduction in payments, the number of active users dropped by 53% within a day. They represented 23% of the total users who had used the system to date. While most users who left reported that they could not invest more time and work for a lower compensation, there were other interesting explanations, such as this:

> "My inbox gets full. I can't find my friend's messages. I sit at my shop, I have time, but now currency is less."

The bonuses seemed to be a hit among active users and the competition element seen before was more pronounced. People gave a missed call even for the slightest delays in receiving messages. One of the earlier lead users said:

> "Everyone has to do 70-80 SMSs now before getting 10 [INR]. So they hope of getting extra from bonuses."

There were other users who had a different point of view:

> "I have free SMS pack, so I just message so that I get 10 Rupees in 1-2 days and I don't go to the shop for recharge. I don't have time to do 200 messages to get leader bonus."

While the bonuses were not sufficient to retain all users, we were curious what fraction of active users stayed because of bonuses. To explore this, we removed the bonuses in the 5th week and looked at users who were active in the last 3 days of the 4th and 5th week. We found that of all the users registered in Phase 1, there were 48 such users by the end of the 5th week, down from 59 in the previous week. This suggests that 19% of those users may have been influenced by bonuses. Interestingly, all 10 active users registered in the previous week sustained their usage in the last week.

RESULTS: SYSTEM PERFORMANCE

To assess the accuracy of mClerk, a random sample of 25 scanned documents containing 1960 words was manually transcribed by a Kannada data entry clerk and verified by a second clerk. Accuracy was defined as the number of words correctly digitized by the system divided by the total number of words. The system achieved an accuracy of 90.1%, which can be considered high given the intermediary transliteration, but is ultimately low according to market standards (details on competitors appear later). Aside from a word's picture clarity on the phone, there are two prominent sources of error. First, two users can enter an equivalent but erroneous English transliteration of a word. Second, the users' responses can be good, but the English to Kannada transliterator can give an incorrect transliteration. To estimate the accuracy of the transliterator, we manually verified whether accepted English responses were valid interpretations of the original Kannada word. In cases in which we deemed the English response to be valid, the transliterator mapped the word correctly in only 95.6% of the cases. This implies that imperfections in the transliterator did have a negative effect on our results.

Still, it proved important to compare responses for equivalence after transliterating them to Kannada, rather than comparing the raw English characters. Of all matching responses, only 75% matched in their English representations. Furthermore, the likelihood of being correct did not differ significantly between replies that matched in English, versus replies that matched following transliteration to Kannada.

Another performance parameter is the average number of checks required per word, before we see an agreement. Of all the digitized words, 63.7% required only two responses, 19.8% required three, 7.9% required four, 4.4% required five, 1.6% required six and the remaining 2.8% required seven or more. All the words eventually reached an agreement. For a word, the two responses in agreement are henceforth referred to as *accepted* responses. On average, 2.58 responses were needed to reach a consensus on each word. Thus, each user needed to submit 1.29 responses (on average) before receiving payment for a word.

The accuracy rate varies with the number of responses needed to reach agreement. This relationship is illustrated in Fig. 4(a). For instance, the point corresponding to 3 responses denotes the number of accurately digitized words which took exactly 3 responses for an agreement divided by the total number of words which took exactly 3 responses

for an agreement. A logistic regression shows that there is a significant effect of the number of responses on the accuracy (Wald statistic = 46.9, $p < 0.0001$). Consequently, these accuracy values can serve as a dynamic confidence value for a word until its responses reach an agreement. We can use this indicator to take actions for low-confidence words, e.g., sending them out for a third verification.

Impact of Word Length

We also investigated ways of predicting the accuracy of a digitization even before the image was sent to users. This would allow us to separate out images predicted to have the lowest accuracies and send them through different, more accurate channels. For instance, if the system is part of a transcription service which also utilizes web-based microtasking, such images could be sent exclusively to the web, avoiding quality degradation and transliteration.

One variable of interest is the word length. Since it is difficult to detect the number of characters in the word (since we have only a scanned image), we report the length in terms of pixels. Prior to distribution, we scaled each image to a height of 28 pixels, while keeping the aspect ratio constant; following scaling, each Kannada character was approximately 14 pixels wide. We refer to the total width of the scaled image as the *word length.* Since a picture SMS supports a maximum length of 74 pixels, words with lengths greater than 74 pixels had to be resized again prior to transmission. To make maximum use of the screen, we did not maintain the aspect ratio while shrinking a word to fit. This resizing and distortion sometimes made it difficult for users to decipher long words.

There is an interesting relationship between the length of a word and the accuracy with which it is digitized in mClerk. Fig. 4(b) depicts accuracy vs. word length. The most visible outlier is for short words (approx. 1-3 characters) for which the accuracy is only 65%. We believe that the errors were partially due to the transliteration algorithm, which achieves an accuracy of only 77.7% for words with lengths 0-40.

A regression test shows that the word length also significantly predicts the number of responses needed to get an agreement (Fig. 4(c)) ($b = 0.004$, $t(1957) = 4.135$, $p < 0.0001$). Therefore, we could identify the images which have a higher probability of needing more responses. For instance, we could set a threshold at 80px and take appropriate action for longer images, such as feeding these into a web-channel as mentioned earlier.

Digitization Latency

The median duration between the time a picture message was sent and the time its response was received was 2.2 minutes (*mean* = 14.9, *SD* = 95.5, *range* = 0.5 – 2622.4). However, requesters would be more interested in the time taken to digitize a word from the point it was first sent to when the second verified response was received. This quantity, the *word digitization latency* as we term it, had a median value of 3.8 minutes (*mean* = 72.4, *SD* = 351.6, *range* = 0.6 – 6010.3). Note that since the usage during the day was very high, it also meant that the server was loaded with high traffic and as a result the turn-around time was at times longer (up to 4 minutes) than low traffic hours in early morning or late night (around 15-30 seconds).

Comparison to Existing Services

To compare our system to the existing state-of-the-art, we hired professional data entry clerks to digitize a subset of our test corpus. On Kannada text, the professional services (detailed in the next section) achieved word-level accuracies of 96.9% and 97.6%. While this is lower than the market standard of 99% for English text [11], it is higher than mClerk's accuracy of 90.1%.

In the future, we envision several techniques to enhance the accuracy of mClerk. First, the system should leverage its knowledge about the skills of individual users. Currently all replies are treated equally, and two matching replies are treated as a correct response. However, some users submit matching replies more frequently than others, and their responses are more likely to be correct. Our data suggests that users behave consistently over time: individual acceptance of a user's replies in the 1st week is correlated with their acceptance in the 2nd week ($r(63) = 0.57$, $p<0.0001$). The system can limit the influence of less reliable workers by matching their response to a trusted worker, or perhaps by requiring a minimum match rate for a worker to qualify for payment. In addition, there is opportunity to introduce additional correctness checks with minimal cost to the system. For example, rather than replicating all tasks, a second user could verify or reject the response of a prior worker. We are confident that with such modifications, the accuracy of mClerk can soon compete with leading market alternatives.

ECONOMIC ANALYSIS

Cost to Requesters

To gain an understanding of the existing market rates for digitizing Kannada text, we conducted a survey of 20 data entry agencies whose contacts were obtained from web search results of multiple queries such as "*Kannada Data Entry*", "*Data Entry Bangalore*", etc.. We requested all the agencies to send us a quote for digitizing Kannada documents in bulk, using a sample from our corpus as guidance. Only 7 agencies replied, out of which 2 provided services exclusively in English: evidence of the difficulty of finding local-language digitization, even from agencies in India. The remaining 5 agencies supplied quotes, which (in INR per word) were 2.8, 2.5, 2, 1 and 1. Expecting a better deal, we negotiated these rates to INR 1.25, 1, 0.8, 1, and 0.5 per word, yielding an average quote of INR 0.91 per word. The lowest priced agency refused to do more than 1500 words per day. We hired the two cheapest agencies for the accuracy experiments described previously.

We also conducted a second survey, which revealed that Kannada data entry is much more expensive than English. In a survey of 19 English transcription companies, 9 replied

with final quotes ranging from INR 0.2 to INR 0.4 per word: roughly three times cheaper than Kannada. This could be due to multiple factors including the ease and speed of English typing as opposed to Kannada. On conversing further with the companies, we found that most companies hire Kannada translators who have to be paid more than data entry workers. Clearly, it is non-trivial to hire Kannada data-entry services for bulk work at reasonable costs and having a quick turn-around time.

mClerk fills this gap with an affordable and highly available digitization service. Taking into account expenses such as referrals and replication, customers of mClerk would have paid INR 1.1 per word in Phase 1, and INR 0.44 per word in Phase 2. In other words, Phase 1 pricing is within 20% of the market rate, and Phase 2 pricing is approximately 2x better than the typical market offering.

Though mClerk pricing is already very competitive, it can also be greatly improved. Currently, up to 40% of the system's payout is lost to the mobile carriers, since many of them charge a staggering 40% service fee on small (INR 10) recharges. That is, while we made a payment of INR 10 into workers' accounts, the amount available for workers' consumption varied between INR 6-10. Workers did not complain about this practice, as it is a well-known aspect of small mobile recharges. However, a scalable business could likely avert this expense via partnerships with carriers. Finally, operating costs can also be decreased by leveraging the same techniques used to increase accuracy. As described earlier, the system could favor reliable workers and issue verification tasks rather than replicating all tasks.

Income of Workers

In conversations with local-language data entry companies, we discovered that full-time clerks are paid between INR 6,000 and 12,000 (USD 133 to 267) per month, depending on the type of data and the location – urban or semi-urban. By comparison, translators at these facilities are paid up to INR 15,000 (USD 333) per month. The basic eligibility criterion is a typing speed of at least 1500 words per hour. Even if digitization speed is 4x slower than typing speed, in a 48-hour work week these clerks are earning at most INR 0.17 per word: more than 5 times less than the price charged to customers. The rest of the costs include computer equipment, work space, personnel management and operational overheads, besides a profit margin which make up for the difference in the quoted price and the amount paid to the worker. mClerk eliminates most of these costs and the benefits go directly to the worker. However, data entry is also slower on a phone than on a computer.

How much can workers earn on mClerk? First, it is important to recognize that we do not intend the system as a substitute for full-time employment. This form of mobile microtasking is appropriate as a source of supplemental income and allows the user to choose the time and duration that they want to work [22]. Moreover, the intention is for the users to source their "working hours" from the small bursts of free time they get during their regular schedules; times in which they otherwise don't have an opportunity to earn. Conservatively assuming that a seamless system maintains an SMS turn-around time of 15 seconds and the time it takes a user to see the message, key in the response and send it is 30 seconds, a user can effectively send in a response every 45 seconds. If the payout in airtime minutes is equivalent to INR 10 per 50 correct messages and a user spends two hours a day doing such tasks, he or she can likely earn up to INR 744 (USD 16.5) as supplemental income in a month (assuming words require 2.6 responses on average). This is about an 8% income increment for the average worker in our study, who otherwise would be unlikely to earn any money during their idle time. In fact, if we assume that we have two server phone numbers with which users exchange messages in alternation, the SMS turn-around time will not affect the user, as it will overlap with their work for the other server. Thus a user can effectively send in a response every 35s, including a 5s window to navigate between messages. A user working for 2 hours per day can earn up to INR 956 (USD 21.2) per month in this optimal scenario.

We note that some may question the ethics of crowd-sourcing of work at low cost in developing regions. Fair pricing of crowdsourcing tasks is an ongoing global debate [18] regardless of developing or developed world contexts. Ross et al. [20] found that Turkers earn under $2.00/hour and MTurk functions as a part-time or even a full-time job for the users. They note that Turkers are positioned as independent contractors and not as employees and are therefore not guaranteed minimum wages. Furthermore, one could argue that since this sort of work is by nature not hazardous by any measure, and workers are free to do the work or decline it at will, fair pricing really comes down to what the market will bear and is not an ethical issue per se. Nevertheless, in the optimal scenario described above, the system offers INR 128 (USD 2.84) for 8 hours of work; this is slightly more than India's National Floor Level Minimum Wage of INR 115 (USD 2.56) per day [10].

DISCUSSION

Several of our findings have implications for the design of future mobile crowdsourcing platforms. We discovered and overcame challenges that affected the users' experience – the multiple-responses problem, inactivity due to the mobile operator dropping messages, partial payments due to the operator's service cuts, message lag during peak hours and saturated resources to handle missed calls. In addition, the system required the users to have free SMS schemes. Some users were even skeptical about the motivation of the system, as it seemed too good to be true.

It is remarkable that despite all of these roadblocks, the usage went beyond expectations. Why was mClerk successful and what can other platforms do to replicate this success? Firstly, users who have occupations that allow them free time as well as social interactions (intermittent or

continuous) are the ideal users. In fact, starting out with users who have a strong social presence will help in ensuring the presence of lead users, who are essential for the system to take-off. Secondly, leaderboards are effective in enhancing the social value of the system and hence, usage. Thirdly, reminders are effective in getting the dormant users to be active. Fourthly, constant feedback on every aspect of the system, such as account status updates and referral earnings is effective in keeping a user engaged.

Independent of mClerk's design, it is also important to understand the fundamental motivations of the users. While earning in free time is certainly a factor, we also saw users putting in dedicated hours. A common theme that emerged from the user interviews was that everyone considered this as a 'service to be activated', just like other services of mobile operators. They thought of our system as a scheme to get daily mobile balance, instead of a part-time job that pays. In that vein, they considered themselves to be taking advantage of a special offer rather than doing work. The social dynamics compounded this further.

One concept worth exploring is whether pitching the system as a social game allows reduced payouts. Taking inspiration from the ESP game [23] that created image metadata by having two users play an online game, we can design our own system which pits two users in a game of word solving. It would be an interesting challenge and experiment to do this over SMS.

CONCLUSION

We believe our work is the first example of crowdsourced work allocation of non-textual tasks for non-English speakers using low-end phones. In conclusion, our contributions are as follows 1) A novel system for mobile crowdsourcing of paper digitization tasks, utilizing SMS-based images to enable participation using low-end mobile phones, 2) The first demonstration of large-scale crowdsourced digitization for a language that lacks font support on workers' devices, 3) An ecologically valid deployment of the system, demonstrating viral propagation through semi-urban communities outside of Bangalore, 4) An assessment of the system's accuracy and performance relative to other solutions for the Kannada language, and 5) Design implications that can inform future mobile interventions in this space.

We argue that our system has the potential to be as accurate and economically viable as other market alternatives. In the future, other contextually appropriate tasks such as audio transcription and tagging locally relevant images and songs might offer the potential for increased payments. We believe that mobile crowdsourcing holds immense potential for emerging markets and our work only scratches the surface of what could become a very powerful ecosystem.

ACKNOWLEDGEMENTS

We are grateful to Nithya Sambasivan, James Davis, Richard T. Guy and Indrani Medhi for helpful conversations and feedback.

REFERENCES

1. txteagle. http://txteagle.com/.
2. MobileWorks. http://www.mobileworks.com/.
3. Samasource. http://samasource.org/.
4. Ushahidi. http://www.ushahidi.com/.
5. Captricity. http://www.captricity.com/.
6. Google Transliteration. http://google.com/transliterate
7. Inside India's CAPTCHA solving economy | ZDNet. http://www.zdnet.com/blog/security/inside-indias-captcha-solving-economy/1835.
8. http://wikipedia.org/wiki/Enhanced_Messaging_Service.
9. http://www.csoft.co.uk/documents/sms3_0_0.pdf.
10. http://pib.nic.in/newsite/PrintRelease.aspx?relid=71533.
11. Ahn, L.V., Maurer, B., McMillen, C., Abraham, D., and Blum, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science, (2008).
12. Alaei, A., Nagabhushan, P., and Pal, U. A benchmark Kannada handwritten document dataset and its segmentation. ICDAR, (2011).
13. Arivazhagan, M., Srinivasan, H., and Srihari, S. A statistical approach to line segmentation in handwritten documents. Proceedings of SPIE, (2007).
14. Dow, S.P. and Klemmer, S.R. Shepherding the Crowd: An Approach to More Creative Crowd Work. CHI EA, (2011).
15. Eagle, N. txteagle: Mobile Crowdsourcing. Internationalization, Design and Global Development, (2009).
16. Frei, B. Paid Crowdsourcing: Current State & Progress towards Mainstream Business Use. Smartsheet White Paper, (2009).
17. Khanna, S., Davis, J., and Thies, W. Evaluating and Improving the Usability of Mechanical Turk for Low-Income Workers in India. ACM DEV, (2010).
18. Norcie G., Ethical and Practical Considerations For Compensation of Crowdsourced Research Participants, CHI WS on Ethics Logs and VideoTape: Ethics in Large Scale Trials & User Generated Content, (2011).
19. Rogers, E.M. Diffusion of Innovations, 5th Edition. Free Press, (2003).
20. Ross, J., Irani, L., Silberman, M., Zaldivar, A., and Tomlinson, B. Who are the crowdworkers?: Shifting demographics in Mechanical Turk. CHI EA, (2010).
21. Sambasivan, N. and Cutrell, E. ViralVCD : Tracing Information-Diffusion Paths with Low Cost Media in Developing Communities. CHI, (2010).
22. Thies, W., Ratan, A., and Davis, J. Paid Crowdsourcing as a Vehicle for Global Development. CHI Workshop on Crowdsourcing and Human Computation, (2011).
23. Von Ahn, L. and Dabbish, L. Labeling images with a computer game. CHI, (2004)