Scaling Requirements Extraction to the Crowd

Experiments with Privacy Policies

Travis D. Breaux and Florian Schaub
Institute for Software Research
Carnegie Mellon University
Pittsburgh, Pennsylvania, United States
{breaux, fschaub}@cs.cmu.edu

***Abstract*—Natural language text sources have increasingly been used to develop new methods and tools for extracting and analyzing requirements. To validate these new approaches, researchers rely on a small number of trained experts to perform a labor-intensive manual analysis of the text. The time and resources needed to conduct manual extraction, however, has limited the size of case studies and thus the generalizability of results. To begin to address this issue, we conducted three experiments to evaluate crowdsourcing a manual requirements extraction task to a larger number of untrained workers. In these experiments, we carefully balance worker payment and overall cost, as well as worker training and data quality to study the feasibility of distributing requirements extraction to the crowd. The task consists of extracting descriptions of data collection, sharing and usage requirements from privacy policies. We present results from two pilot studies and a third experiment to justify applying a task decomposition approach to requirements extraction. Our contributions include the task decomposition workflow and three metrics for measuring worker performance. The final evaluation shows a 60% reduction in the cost of manual extraction with a 16% increase in extraction coverage.**

***Index Terms*—requirements extraction; natural language; crowdsourcing**

I. INTRODUCTION

Requirements engineering research has long dealt with the challenge of processing natural language texts to facilitate communication, specify systems and conduct requirements validation [34]. Recently, new sources for requirements extraction and analysis have come to include project-specific mailing lists [22], privacy policies [7, 29] and regulatory codes and laws [11, 16], to name a few. The challenge of acquiring requirements from such sources, however, is that each medium carries a native writing style that proficient readers are accustomed to: for example, the style of exchanging e-mails is similar to writing professional letters in which actors interact through text, defining important terms, critiquing positions and adding clarficiation [22]; whereas, the style of law is more structured with preambles, definitions, sections based on topic and cross-references [5].

Requirements researchers have extracted requirements from text using a combination of manual and automated approaches. Manual approaches may include simple, repeatable steps accompanied by a coding frame that are used to classify the text [35]. To assess validity, researchers apply various forms of inter-rater reliability to coded data sets extracted by multiple, trained analysts. The value of manual extraction is that, when the method is derived from the dataset, called a *grounded theory* [12], the derivation process surfaces complex nuances and boundary cases that are more likely addressable using human-interpretable heuristics. Such boundary cases are often difficult to address using automated approaches. In prior work, for example, Breaux et al. discovered heuristics for inferring *implied* rights and obligations from explicitly stated requirements to increase requirements coverage [8]. However, the downside of manual methods is the challenge of scalability: achieving a two-fold increase in the number of documents processed requires considerable effort by a small number of expert analysts. Moreover, automated approaches, including machine learning, rely on large data sets to evaluate performance in cross-validation studies [3]. Overcoming this challenge of scaling manual extraction could lead to new analytics that leverage unprecedentedly large datasets.

Crowdsourcing and human computation provide a middle ground between manual extraction by a few experts and natural language processing. Crowdsourcing has emerged as a viable approach for leveraging human intelligence – often provided by non-experts – in the context of problems that remain hard to solve with automated methods [33], such as information extraction from noisy images [1], politicial sentiment analysis [18], and text translation [4, 38]. In crowdsourcing, tasks are typically divided into smaller, more managable *microtasks*. These microtasks are then distributed to the *crowd* – a large number of independent workers who offer their services using crowdsourcing platforms, such as Amazon Mechanical Turk[1] (MTurk) or CrowdFlower. [2] Crowdsourcing results are subsequently combined to yield a solution for the larger task. Major challenges in crowdsourcing complex tasks include designing a task workflow that produces high quality results, while keeping per-task costs within budgetary constraints.

In this paper, we report results from three experiments aimed at assessing the potential of crowdsourcing requirements extraction to non-experts. We based our experiments on a recently published method for manually extracting requirements from privacy policies [7] as follows: the first experiment assessed the crowd workers' ability to apply a coding frame, and measured consensus building across workers; the second experiment assessed the ability of crowd workers to extract complete requirements using multiple

[1] https://www.mturk.com
[2] https://crowdflower.com

semantic roles, simultaneously; based on the gained insights from these two experiments, a third experiment assessed a task decomposition approach, in which semantic role labeling was devided into separate, interdependent microtasks to minimize worker effort and increase data quality. We found that our approach yields nearly a 1.5:1 cost decrease and increased requirements coverage compared to manual extraction by trained experts. The contributions of this approach include a validated task decomposition workflow and generalizable metrics for worker performance, which we believe can be used to crowdsource similar requirements extraction tasks.

The remainder of this paper is organized as follows: in Section II we review related work on designing workflows for crowdsourcing complex tasks; in Section III we introduce the manual extraction method for privacy requirements and corresponding economics based on two prior case studies; in Section IV, we present the results of two preliminary crowdsourcing experiments based on the manual extraction method; in Section V, we present our task decomposition approach, experiment and our main results. A discussion follows in Section VI with the conclusion in Section VII.

II. Designing Crowdsourcing Workflows

Crowdsourcing has been succesfully applied to many different tasks and contexts [13]. Snow et al. [36] showed that crowdsourced annotations from non-experts reach a high level of agreement with expert annotators for different natural language annotation tasks, such as word similarity, word sense disambiguation and textual entailment recognition. We now review related work on data quality and task design.

A. Data Quality

Annotation quality is impacted by crowd worker and task characteristics [2]. Workers are typically non-experts of the task, which may introduce inaccuracies and noise in results. Dishonest workers may try to cheat to obtain payment. Complexity of tasks and instructions of a task can further induce high cognitive demand [15], which may reduce result quality [2]. Thus, crowdsourcing annotation tasks should be understood as a noisy classification process. A number of quality control strategies have been proposed [33] at design or runtime to ensure higher quality results [2]. We now discuss data quality approaches relevant to requirements extraction [2, 33].

Pre-screening and reputation. Workers can be asked pre-screening questions to evaluate their suitability before assigning tasks [32]. A certain reputation level, e.g., 95% accepted submissions, can be used to target high-performing workers, which has been shown to signficiantly influence result quality for low pay tasks [23].

Review and quality control. After tasks have been submitted by workers, results can be reviewed before accepting them. If a submission is rejected, pay is withheld for that task. Expert review is an expensive approach that limits scalability, because domain experts must review and judge the quality of each submission before accepting them [2]. Multi-level review, on the other hand, leverages a second group of crowd workers to verify data annotations provided by the first group [33]. Automated verification can be used for tasks for which verifying result correctness is easier than its computation [33].

Incentive structure and economic model. Incentive structures can be optimized to obtain higher data quality at lower costs [33]. Provided monetary incentives can impact data quality. Kazai finds that higher paying tasks lead to good worker performance regardless of the qualification [23]. However, she also finds evidence for diminishing returns as workers who are paid low wages report to be happier with their compensation than highly paid workers. Thus, higher pay does not necessarily result in higher quality [2]. Horton et al. desribe a labor economics model of crowdsourcing [17]. Their model is based on a reservation wage, i.e., the lowest compensation a worker is willing to accept for a task. In experiments, they determine that the reservation wage is log normally distributed with a median wage of $1.38/hour.

Redundancy and majority consensus. To improve reliability with noisy data, the same task can be assigned to multiple workers in order to determine output agreement. Output agreement can be measured by majority voting or weighted voting, which takes into account worker performance [33]. A large variety of crowd consensus approaches have been proposed [19], which also resulted in benchmarks to compare their effectiveness [19, 37]. Redundancy, which is needed in all consensus based approaches, may entail higher costs [20].

Identification of low performers. Result quality can be improved (or redundancy reduced) by detecting low-performing or potentially dishonest workers. An effective approach is ground truth seeding [33], in which microtasks for which gold standard data exists are mixed into a worker's assigned task. Failing those tasks can indicate low performance and potentially dishonest intent. At the same time, such verifiable tasks can be used to provide training to workers before they engage in tasks for which no gold standard data exists [24]. Signaling to users that responses will be scrutinized further encourages honest task completion [24]. Statistical filtering can further be employed to determine how far workers deviate from the majority [20, 33]. Once low performers are identified, their results can be rejected and they can be excluded from future assignments, or they can be provided with shepherding and real-time support to improve their performance [2]. Kamar et al. [21] propose a method for routing tasks to workers that optimizes hiring decisions for maximium utility. Based on previously collected task repsonses as well as knowledge about individual worker performance, they estimate what value additional task submissions would provide to the result.

Defensive task design. Carefully phrasing tasks and associated instructions can reduce ambiguities and thus improve result quality [33]. Kittur et al. [24] note that tasks should be designed so that cheating is no easier than completing the task honestly, i.e., creating believable invalid responses should be as effortful as providing honest responses. Careful arrangement of task order can also increase quality.

While data quality can be mesured by accuracy, i.e., the ratio of correct labels to the total number of labels, whether a label is correct is often not objectively verifiable. Hsueh et al. propose practical aspects to be considered by quality metrics [18]: *annotator-level noise* reflects an annotator's accumulated noise level across tasks, whereby noise is defined as a deviation from the majority vote; *item-level ambiguity* reflects that different items are not equally easy to annotate; *inherent* *lexical uncertainty* of items can be determined by the uncertainty of an automated classifier for the respective item. Allahbakhsh et al. [2] further note that the subjective nature of quality and the fact that many quality control methods are tailored to specific task domains make it difficult to compare quality metrics.

B. *Task Design and Decomposition*

Robust task design is an esstential aspect of crowdsourcing complex tasks. Instruction length and complexity can increase cognitive demand and thus decrease worker performance [15]. Complex payment schemes also increase cognitive demand, because workers focus on optimizing their payment in addition to completing the task [15]. Thus, task design that channels cognitive demand towards the tasks is potentially more effective. Repeating tasks also facilitates a learning effect for workers which reduces cognitive load and thus potential for error [15]. A related aspect is that the usability of the crowdsourcing interface impacts result quality [2]. Eickhoff and de Vries [14] further find that phrasing tasks in a non-repetitive manner discourages automation on the worker side and yields higher quality results.

Allahbakhsh et al. [2] note that task granularity affects data quality. They distinguish between simple tasks that are self-contained and require low expertise and complex tasks. Complex tasks should be decomposed into simpler microtasks, of which the results can be later consolidated to solve the larger task. Microtasks can be chained to form iterative and parallelized task workflows [2]. TurKit [28] allows to integrate MTurk tasks into program code with the purpose of automating human computation with complex workflows. Kittur et al. propose CrowdForge as a general purpose framework for solving complex and dependent tasks with a map-reduce approach [25]. CrowdForge supports dynamic multi-level partitioning so that workers themselves can decide how to subdivide a task, which then generates new microtasks automatically. Result aggregation can be automated or partially performed by workers. Turkomatic [26] further aims to crowdsource the actual design of a task workflow. Workers decompose tasks on their own while the requester can modify workflows in realtime. Cascade [10] is an automated workflow for creating object taxonimies. Cascade's task decomposition is based on three task primitives: *generate label* (show multiple items, ask to find category), *select best label* (given a single item and multiple labels, select the best fit), and *categorize* (for a single item and multiple categories, vote for each category). The motivation is that these primitive tasks can be completed in 20 seconds and highly parallelized.

Zaidan and Callison-Burch [38] find that an optimized task workflow results in near-professional quality for natural language translations. In their process, statements are first translated redundantly, those translations are then redundantly edited for fluidity by native speakers, and then those edits are again ranked by crowd workers. Based on a quality metric that incorporates the ranking score, as well as sentence and worker features, best fits are selected automatically. Ambati et al. [4] also propose a crowdsourcing workflow for translation tasks in which results are enhanced through multiple levels, including lexical translation, assistive translation, and monolingual synthesis to refine translated sentences. Negri et al. [31] employ task decomposition to build a cross-lingual entailment corpus with crowdsourcing. Their workflow consists of sentence modification, annotation, and translation tasks. They find that the resulting workflow can be scaled easily for a large set of original sentences.

C. *Crowdsourcing and Requirements Engineering*

Requirements engineering research is beginning to leverage crowdsourcing in elicitation. Lim and Finkelstein [27] introduce the StakeRare method that employs social networks and collaborative filtering to elicit and prioritize user requirements. The approach was applied to a 30,000 user system with 87 stakeholders and showed improved completeness and accuracy in predicting stakeholder priorities. Caire et al. [9] employ crowdsourcing to elicit visual notations from novices and study their utility. In our work, we investigated task decomposition for requirements extraction in multiple crowdsourcing experiments with the goal of obtaining an optimized and scalable task worflow. Our work centered on the extraction of privacy requirements from privacy policies.

III. Manual Extraction Method

Our three experiments are based on a manual method for extracting privacy requirments from privacy policies [7]. In the manual method, the analyst first classifies statements using a statement-level coding frame based on action type (e.g., does it refer to collection, use or transfer of personal information). Next, the analyst applies a phrase-level coding frame to identify the information type, recipients or senders of information, and purposes for which information is used. The complete coding frame is presented in Appendix A. In Fig. 1, we present an example policy statement that has been manually coded in a prior case study [7]: phrases are highlighted corresponding to codes for words indicating modality, transfer, datum, target, etc. In the complete manual method, the extracted phrases are mapped to logical formula, which are then used to reason about conflicts between permitted and prohibited actions [7]. In this paper, we only treat the extraction problem, and leave crowdsourcing of translation to logic to future work.

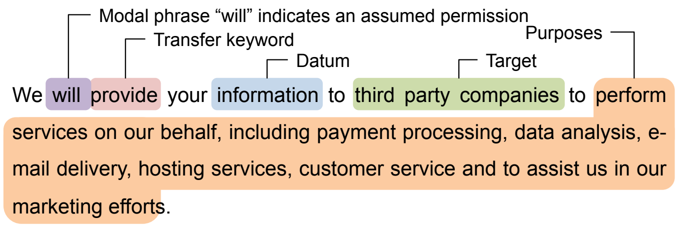

Modal phrase "will" indicates an assumed permission
Transfer keyword
Datum
Target
Purposes
We will provide your information to third party companies to perform services on our behalf, including payment processing, data analysis, e-mail delivery, hosting services, customer service and to assist us in our marketing efforts.

Fig. 1. Example coded policy statement: the stateement-level code "Transfer" is mapped to the action verb "provide," the phrase-level codes identify the modal phrase, datum, target and list of purposes.

There are multiple factors to consider when evaluating how well crowdsourcing can contribute to scaling this task, including the time to extract a requirement, the cost of extraction and the proportion of extracted statements to the size of the policies. In our prior study [7], our analysts spent between 1.4 and 1.8 minutes per policy statement coding the policy text. The cost to code the Zynga privacy policy, which consisted of 189 statements, at a pay rate of $15 per hour would be $86.25 or about $0.46 per statement. To assess validity, we use the inter-rater reliability Kappa statistic, which requires multiple analysts and which at least doubles the cost to code the Zynga policy to $172.50. This cost and effort includes the time to read the policy, identify relevant statements, and assign the corresponding codes; it does not include the time to translate the coded phrases into logic or a requirements specification language. In addition, we found that the analysts coded only 28-49% of the policy statements. Many uncoded statements reside in specific sections of the policies that can be eliminated to reduce the cost of the manual effort as well.

In the next section, we describe two pilot experiments and our results to adapt this method to the MTurk crowdsourcing platform. In the third experiment, we discuss how the pilot studies and cost of the manual approach factored into our task decompostion design choices.

IV. Preliminary Experiments

The task designs for the first two experiments E1 and E2 aimed at answering two research questions: how well do crowd workers apply the statement-level and phrase-level coding frames, and how many crowd workers are needed to reach consensus? We now discuss the two designs and results.

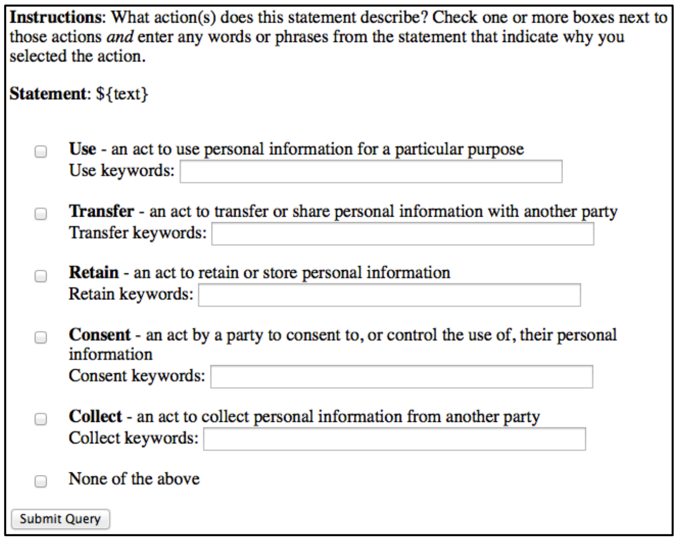
A. *Sentence Classification and Consensus Building*

Experiment E1 evaluates the crowd workers' ability to apply the sentence-level coding frame (see Appendix A) and to measure how many crowd workers are needed to reach consensus. For this task, we created a stratified sample of nine sentences selected from our prior case study [7], which we use as a gold standard to evaluate the crowd worker results. We selected sentences along the following strata: *single-coded*, in which only a single sentence-level code applies; *dual-coded*, in which exactly two sentence-level codes apply; and *none*, wherein no codes apply. We chose this strata to ensure coverage across the coding frame and assess how well workers respond to statements with multiple codes. The selected sentences, prefixed by the expected codes in bold, are as follows:

1. **Collect**: "We may collect or receive information from other sources including (i) other Zynga users who choose to upload their email contacts; and (ii) third party information providers."
2. **Consent, Transfer:** "We do not actively share personal information with third party advertisers for their direct marketing purposes unless you give Us your consent."
3. **Retain**: "Zynga stores information about site visitors and players on servers located in the United States."
4. **Collect, Retain**: "We receive and store the information you provide, including your telephone number, when you sign up to have SMS notifications sent directly to your mobile phone."
5. **Transfer**: "To properly credit user accounts and to prevent fraud, a unique identifier, in some cases your user ID number, will be shared with the offer wall provider.[3]"
6. **Use**: "This information will be used to supplement your profile – primarily to help you and your friends connect."
7. **Collect, Use**: "The information collected may be used to offer you targeted ad-selection and delivery in order to personalize your user experience by ensuring that advertisements for products and services you see will appeal to you, a practice known as behavioral advertising, and to undertake web analytics (i.e. to analyze traffic and other end user activity to improve your experience)."
8. **None**: "Zynga implements reasonable security measures to protect the security of your information both online and offline, and We are committed to the protection of customer information."
9. **None**: "Zynga games or their purchase pages may display an 'offer wall' that is hosted by an offer wall provider."

[3] An offer wall is a web page or screen with multiple offers aimed at providing users more choices for clicking on ads.

Figure 2 shows the task interface for E1, in which the variable ${text} would be replaced by one of the statements above. The instructions are simple with no examples. Response options (use, transfer, etc.) were always presented in random order with "none of the above" always appearing last. Workers may select as many categories as desired, but they must provide phrases to justify their answers.


Instructions: What action(s) does this statement describe? Check one or more boxes next to those actions *and* enter any words or phrases from the statement that indicate why you selected the action.

Statement: ${text}

- [ ] **Use** - an act to use personal information for a particular purpose
  Use keywords:
- [ ] **Transfer** - an act to transfer or share personal information with another party
  Transfer keywords:
- [ ] **Retain** - an act to retain or store personal information
  Retain keywords:
- [ ] **Consent** - an act by a party to consent to, or control the use of, their personal information
  Consent keywords:
- [ ] **Collect** - an act to collect personal information from another party
  Collect keywords:
- [ ] None of the above

Fig. 2. User interface to the sentence clasification task; the variable ${text} is replaced by one of nine sentences; workers are asked to provide matching keywords from the text to justify their responses.

For E1, we recruited US residents as workers on MTurk, who had at least a 95% approval rating for over 5,000 tasks. We paid workers $0.15 per sentence classification task and allowed up to 20 minutes to complete the task. Results were accepted or rejected within 24 hours. On average, workers required 66 seconds to complete a single sentence classification that resulted in an average hourly rate of $8.18.

In response to our request, we solicited 50 workers per task that resulted in a total of 76 distinct workers participating who classified between 1-9 sentences each to yield 448 classifications (i.e., workers are not required to complete all tasks). Table I presents the results as the number of ratings per sentence (S#) by category: cells shaded dark blue represent majority consensus (i.e., $\geq 25/50$ workers assigned the category to this sentence) and cells shaded light orange represent near majority consensus or NMC (i.e., 13-24 workers assigned the category to this sentence). The dark blue-shaded cells match our expected results with 100% precision and recall. After inspecting the results, we believe NMC is due to implied actions perceived by some workers, e.g., the action of "signing up" for a game account (statement 4) implies that the user will transfer some personal information to the game website.

TABLE I. FREQUENCIES OF CODES PER SENTENCE

S#	Collect	Consent	Use	Retain	Transfer	None
1	48	8	9	19	15	0
2	8	46	21	7	38	0
3	12	0	2	49	3	0
4	38	16	13	44	8	0
5	7	2	22	6	40	3
6	13	4	42	12	10	1
7	36	5	44	10	10	0
8	7	3	5	19	0	25
9	5	2	8	1	8	31

Due to space limitations, Fig. 3 only presents plots for the percentage of crowd workers who classified even sentences 2, 4, 6, 8: the y-axis shows the percentage of workers who assigned the category, and x-axis shows the response from the i^{th} worker from the 1st to the 50th in time order. Across all 9 plots, we observe that percentages converge at between 5-15 workers per statement, which indicates that higher worker numbers are not necessary. We used this information to set the number of invited workers to complete experiments E2 and E3.

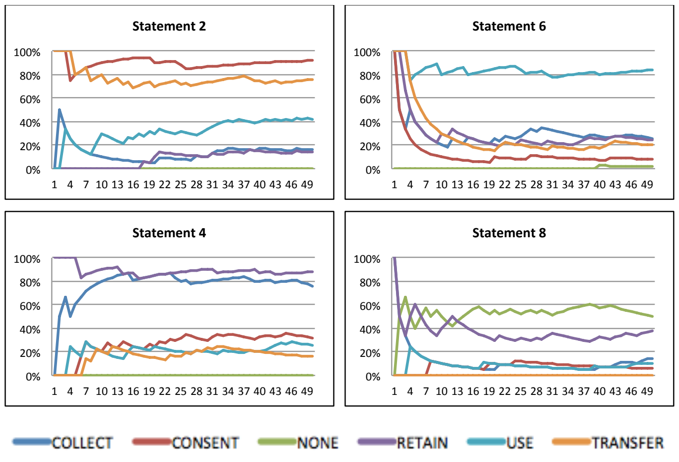
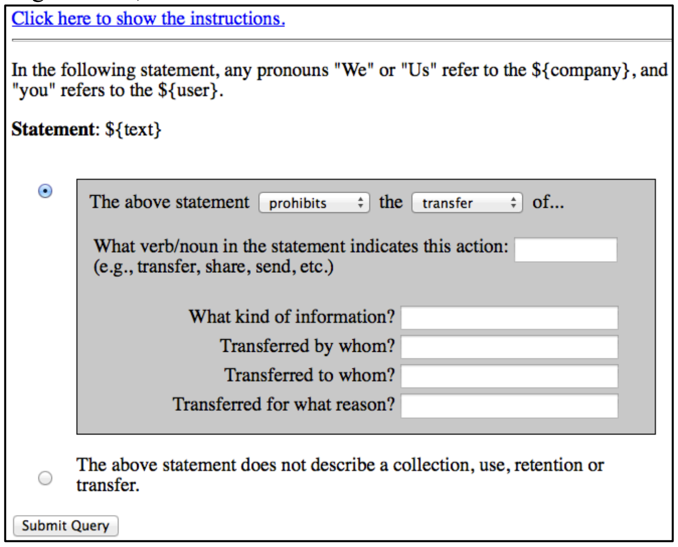


Fig. 3. Percentage of crowd workers who labeled statements by concept; the y-axis is the percentage, and the x-axis is the worker number from the 1st worker to the 50th worker; the proportions stabilize between 5-15 workers.

We evaluated worker performance in E1 by their ability to assign ratings within the majority consensus (i.e., of all their ratings assigned, what proportion match the majority rating). We found for each task that 12/50 workers provided between 40-50% of their ratings within the majority, and 18/50 workers rated with the majority less than 50% of the time. In addition, all workers did not complete all tasks: 16/50 workers attempted only one sentence classification task, whereas 32/50 workers completed all nine tasks. We computed the average number of tasks completed by the worker's quartile ranking with respect to majority ratings and found that Q1=1.9, Q2=5.0, Q3=6.6, and Q4=7.2, which suggests workers rating in the majority are more likely to complete more sentence classification tasks.

B. *Full-frame Extraction*

Experiment E2 is a "full-frame" extraction task that combines the sentence-level coding frame with the phrase-level frame described in Appendix A. Workers are shown one sentence and they are asked to answer up to seven questions by completing an online form (see Fig. 4). The questions ask: what is the *modality* (permit, prohibit), what is the *action* (collect, use, transfer, retain), what is the *indicative verb* for this action, what *kind of information* is acted upon (the datum), *to whom* is information transferred (target), *from whom* is information collected (source), and *for what reason* or purpose is the action performed. Alternatively, the worker can indicate that the sentence does not describe any of these actions. We further include clickable instructions that expand to show a working example. The action names listed by the verb textbox are selected from the top three verbs for each action identified in our Zynga case study [7]; thus, "transfer, share, send" were the three most commonly observed verbs indicative of a transfer action. The last four questions appear dynamically based on what action has been selected in the drop-down list. For this task, we selected 18 new sentences from the Zynga case study [7] using the same stratification criteria as in E1: single-coded, double-coded and none.


Click here to show the instructions.

In the following statement, any pronouns "We" or "Us" refer to the ${company}, and "you" refers to the ${user}.

Statement: ${text}

[✓] The above statement prohibits the transfer of...

What verb/noun in the statement indicates this action: (e.g., transfer, share, send, etc.)

What kind of information?

Transferred by whom?

Transferred to whom?

Transferred for what reason?

The above statement does not describe a collection, use, retention or transfer.

Submit Query

Fig. 4. User interface to the full-frame extraction task.

We solicited 15 workers per sentence based on E1 results that show sentence-level classification consensus converges between 5-15 workers. These workers were US residents and they had at least a 95% approval rating for over 5,000 tasks. We paid workers $0.15 per full-frame extraction and allowed up to 10 minutes for workers to complete the task. Results were accepted or rejected within 48 hours. For E2, we received 38 workers who classified between 1-18 sentences yielding 135 classifications. The average task completion time was 89 seconds with an average hourly rate of $6.07.

We believe the E2 task was more challenging for the workers than the E1 task. In addition to the longer average time required to complete E2 tasks, we saw that individual workers completed fewer tasks in E2 than in E1. Furthermore, the data quality produced by E2 workers is more varied. Figure 5 presents the number of perfect responses received along the y-axis and the sentence number along the x-axis, respectively. For each column, we report the average worker score for each sentence, which is computed as a ratio of correct answers per total number of possible responses (e.g., for sentence 1, 92% of the questions asked about this sentence were correctly answered across all 15 respondents; a worker's response is *perfect* if it has a score of 100%).

From these results, we drew a few observations. First, the darker shaded, blue columns show that more than two-thirds of workers produced perfect scores for the respective sentence. In most of these cases, the average score is above 90%. Second, sentences with medium shaded, red columns exhibit an above average score of 77% or greater, but a relatively low number of perfect responses. This is because the cumulative number of correct answers across all fifteen responses is high, whereas few workers could correctly answer all questions. The lighter shaded, orange columns correspond to sentences for which respondents were equally divided between providing answers or selecting the negative response option, which was the correct answer. This may indicate *acquiesence bias* [30], in which respondents feel compelled to complete the template, despite encountering a sentence that is not clearly classifiable.

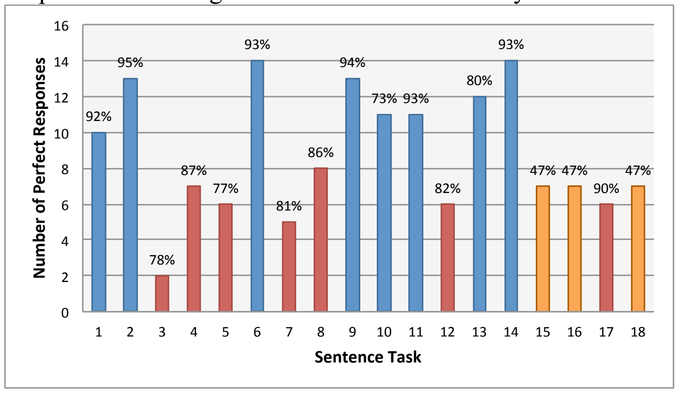


Fig. 5. Number of perfect responses for sentences; percentages correspond to average worker score for each sentence.

While the results from E2 are promising, overall, scaling this full-frame extraction task would be too costly. Recall, the cost for two expert analysts to encode the Zynga policy, which contains 189 statements, is $172.50. With respect to E2, hiring 15 workers to apply the full-frame extraction to the whole Zynga policy at $0.15 would cost $425.25. Therefore, we had to further reduce the number of workers needed and/or increase the quantity of information processed to reduce the overall cost. Therefore, we are inclined to separate the questions in the full-frame extraction into separate tasks, while increasing the quantity of text per task. The idea being that we can recombine the answers in a map-reduce style human computation.

In addition, the E2 results suggest two hypotheses. First, one class of sentences yields perfect responses among two-thirds of respondents (see blue columns in Fig. 5): if we only hire five workers, more than half will produce consensus and perfect responses. Second, a different class of sentences show that only one-third of workers produce perfect responses (see red columns in Fig. 5), but three-fourths of aggregate worker answers in those responses matched consensus with the one-third who provided perfect responses. These hypotheses informed the experimental design of E3 (see Section V), including the development of three metrics to evaluate which workers produce higher quality responses.

V. TASK DECOMPOSITION FOR EXTRACTION

The results from experiment E2 demonstrate several trade-offs in using untrained crowd workers: (1) combining multiple, hetereogenous microtasks into a larger task increases cognitive demand, which necessitates an increased payment to cover the additional time spent by the worker; and (2) consensus is reached earlier on some microtasks than others. Based on these observations, we designed a task workflow in which the full-frame extraction task from E2 is decomposed by semantic roles into microtasks and distributed to multiple workers. The task decomposition design is driven by assumptions derived from the crowdsourcing literature and our experience as follows:

A1. Extraction of complete requirements may require the context of multiple sentences or regions of text;

A2. Once a worker understands a simple task, they can more effectively perform it repeatedly compared to switching between different tasks during the same period [15];

A3. Certain steps in the extraction process can be performed by automated NLP with acceptable levels of precision and recall [36];

A4. Certain microtasks depend on the results from other microtasks, thus suggesting natural break points and ordering of microtasks in a task decomposition;

A5. The financial cost of a task is directly proportional to the task complexity [2, 33] or cognitive demand, thus decomposition should coincide with a smaller cost per microtask, but not necessarily a smaller overall cost.

We designed the task workflow for experiment E3 under these assumptions as follows. First, we chose to provide workers with complete paragraphs, in contrast to E1 and E2 that presented only single sentences. A paragraph provides additional context that may be needed to extract a complete requirement (A1) and we can leverage improved worker performance by having them repeat the same, smaller task for the same duration of 60-90 seconds over larger portions of text (A2). Second, we observed that some microtasks can be partially solved by automated NLP (A3), thus, reducing the workload. Next, some questions need not be answered, if a previous microtask provided a negative result (A4), e.g., we do not need to ask "from whom" information is collected, if the text does not describe collection. These two assumptions A3 and A4 motivate our expectation that we can reduce the cost by skipping portions of policies. The challenge is maintaining validity by ensuring each question is answered by a sufficient numebr of workers to reach consensus.

Figure 6 presents our task decomposition that follows this line of reasoning. Preliminary microtasks that have not been discussed so far, including downloading privacy policies and extracting paragraphs, are shown and have been realized as semi-automated tasks. The microtask "identify modal verbs," such as *may*, *will*, *may not*, etc. is automated. Microtask 4 (*identify the action verbs*) is the lead crowdsourcing task: this task is used to evaluate worker performance early based on a list of known action verbs.; if actions are detected by workers, microtasks 5-7 are created for those paragraphs. Microtask 6 is restricted to paragraphs describing collections or transfers, and actions coded by a majority (>2 raters) are highlighted in this microtask. The results from microtask 5 and 6 are compared with microtask 4 (dotted lines) to assess worker quality.

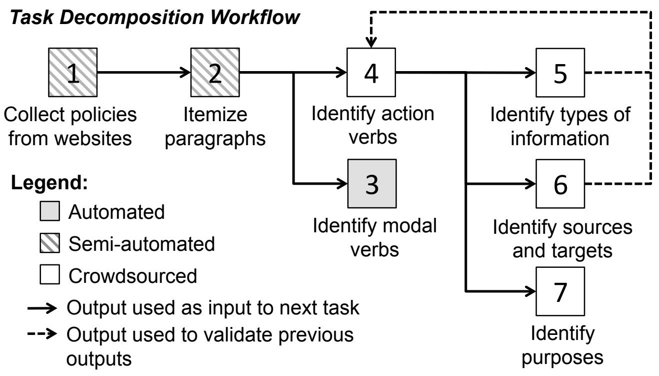

Task Decomposition Workflow
1
Collect policies from websites
2
Itemize paragraphs
4
Identify action verbs
3
Identify modal verbs
5
Identify types of information
6
Identify sources and targets
7
Identify purposes
Legend:
Automated
Semi-automated
Crowdsourced
Output used as input to next task
Output used to validate previous outputs

Fig. 6. An overview of the task decomposition workflow.

To effectively support workers in microtask 5, we developed a specialized user interface for coding key phrases (see Fig. 7). In the interface, users select relevant phrases and press "concept" keys to code the phrase with a specific concept. Thus, workers can highlight phrases as they read through the text, as opposed to copying/pasting or typing answers, which can lead to typographic errors that prolong the task and potentially reduce workers' hourly wages.

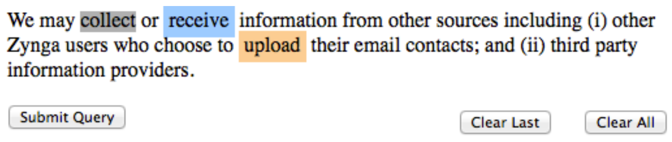

We may collect or receive information from other sources including (i) other Zynga users who choose to upload their email contacts; and (ii) third party information providers.

Fig. 7. Example interface used in task decomposition, step 5.

In the experiment E3, we ask workers to complete steps 4-7 of our task workflow for the following five privacy policies: Zynga.com, Rovio.com, Amazon.com, Walmart.com, and Waze.com. We previously studied Zynga's and Waze's privacy policies, which provides a gold standard by which to compare crowd worker results with our manual method. We selected three additional policies to complement this selection: Rovio's policy covers the popular game Angry Birds, which complements Zynga's Farmville game; Amazon and Walmart are both large retail companies, wherein Walmart specializes in brick-and-mortar sales and Amazon specializes in online sales.

In preparation, we downloaded the policies from their websites, itemized the paragraphs and removed sections describing the following content: the introduction, "contact us," security, U.S. Safe Harbor, policy changes, and California citizen rights. This content is often separated by section or detected with common keywords. In addition, we manually split paragraphs to yield text spans less than 120 words, noting that we preserved some large spans when anaphora referred back to a previous sentence (e.g., when "This information…" depends on a prior sentence to understand *which information* is referred to). We approximate the time to divide these policies to be about 15 minutes per policy, which adds an incidental one-time cost to each workflow instance. Table II summarizes the E3 dataset, including the total policy sections, sentences count, percent of retained sentences after removing sections, and total number of assignments created per policy. Each assignment is a separate microtask instance in the workflow.

TABLE II. SUMMARY OF POLICIES STUDIED IN EXPERIMENT E3

Policy	Sections	Sentences	Retained	Assign.
Amazon	16	95	69.5%	18
Rovio	13	84	83.3%	18
Walmart	28	169	71.0%	27
Waze	17	135	85.7%	34
Zynga	36	195	69.2%	32

We developed a grading system to evaluate worker performance in experiment E3. We cannot use inter-rater reliability to evaluate consensus for phrase-level coding because the non-coded phrases dominate coded phrases and workers generally agree about which phrases not to code, which causes Fleiss' Kappa to converge at 0.99 for all tasks. Thus, we compute a *worker grade* based on the distribution of their ratings among rater agreement as follows:

- Grade 4, if $\geq$ 50% of their ratings shared by 5 workers
- Grade 3, if $\geq$ 50% of their ratings shared by $\geq$ 4 workers
- Grade 2, if $\geq$ 50% of their ratings shared by $\geq$ 3 workers
- Grade 1, if $\geq$ 50% of their ratings shared by $\leq$ 2 workers

In addition, we compute a *ratings/task measure* that describes the ratio of total ratings to total tasks completed by workers. These two measures balance each other: a worker may submit one "safe" rating per assignment that five workers agree with to earn a Grade 4, whereas a worker who submits more than eight ratings per task may be more likely to yield increased coverage with fewer workers in agreement and earn a Grade 1.

Consider Table III as a sample data set for computing worker grades: the *Workers* column lists a unique index for each of three workers W1-W3; the *Ratings* column lists worker ratings in three ranks R1-R3 where a rating is counted in only one column depending on how many workers agree with the rating (e.g., a rating counted under R3 means exactly three workers assigned the same rating to the same item); the *Rank Proportion* columns list the proportion of worker ratings that fall into one of the three ranks (e.g., W1 produced 60/90 ratings in rank R1, so they have proportion P1 = 66%). Finally, grades are assigned based on the rank proportions, e.g., worker W2 received a grade of 3, since $\geq$ *50%* of their ratings (50/90 under R3) are shared by three or more workers (P3 = .556).

TABLE III. EXAMPLE RANKING AND GRADING COMPUTATION

Worker	Ranks			Rank Proportion			Grade
	R1	R2	R3	P1	P2	P3	
W1	60	20	10	0.667	0.250	0.125	1
W2	15	25	50	0.167	0.278	0.556	3
W3	20	40	30	0.250	0.444	0.333	2

Finally, we compute a *sanity measure* after microtask 4 to check worker performance against our predicted minimum level of performance. From our prior case study [7], we compiled a list of the 11 most frequently coded action verbs and we use these verbs in three tenses (past, present, present continuous tense) to predict the classification of actions in each paragraph at microtask 4. For each worker who completed this microtask, we compute the sanitfy measure as the ratio of known actions identified by the worker to known actions predicted in the paragraph. Together, these three measures help us identify workers that contribute different kinds of quality to the results, which we discuss in Section V-A.

A. Results of Task Decomposition Experiment

Table IV presents summary statistics per policy for the four microtasks in experiment E3. We recruited workers who are US residents and who had at least a 97% approval rating for over 5,000 tasks. We paid workers \$0.10 for microtasks 4-5, and we paid \$0.08 for microtasks 6-7, due to their lower complexity. For all microtasks, workers were allowed up to 10 minutes to complete each task. Results were accepted or rejected within 48 hours. For all microtasks, we rejected worker results only if either: (1) the worker produced no codes and 4/5 other workers submitted at least one code for the assignment, demonstrating viable answers missed by the worker; or (2) the worker misunderstood the instructions and coded the wrong microtask (e.g., for information types, they coded actions). The second condition occurred once in microtask 5 and 6, each time with a different worker. When a result was rejected, it was sent back to the platform and another worker was permitted to complete the assignment.

TABLE IV. SUMMARY OF TASKS IN EXPERIMENT E3

By Policy	Total Assigns.	Total Rejects	Avg. Time	Avg. Pay Rate
Amazon	385	25	63.8s	\$5.18
Rovio	381	21	53.8s	\$6.10
Walmart	571	31	49.3s	\$6.59
Waze	732	52	50.3s	\$6.54
Zynga	649	9	63.8s	\$6.18
By microtask	**Total Assigns.**	**Total Rejects**	**Avg. Time**	**Avg. Pay Rate**
4: Actions	655	10	54.0s	\$6.70
5: Information	715	70	50.4s	\$7.25
6: Sources/Targets	703	58	58.6s	\$5.04
7: Purposes	645	0	52.8s	\$5.49

Figure 8 presents the total ratings for the six highest-yield workers who produced the most action ratings on the Zynga policy. The lowest grade worker V1, produced the most ratings (i.e., >75) unreported by any other worker. The highest grade workers V3-V6 produced fewer ratings, but generally produced ratings that many other workers agreed with. Across the entire data set, the average ratings per task for grades 1 through 4 were, respectively: 8.01, 7.74, 5.24, and 4.02.

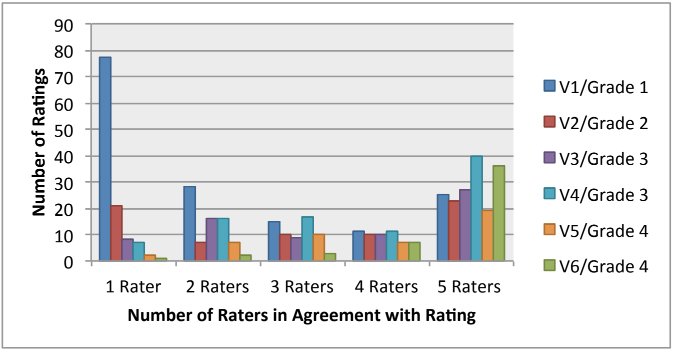


Fig. 8. The six workers with the highest number of attempted assignments for the Zynga policy; workers with more ratings/assignment typically have lower grades, because they find phrases coded by fewer workers.

In the task decomposition, we use a list of known actions from our prior case study [7] to evaluate worker performance. Table V presents the overall results for predicted responses: for each *Policy*, we show the number of verbs predicted as a collection (C), use (U), transfer (T), retention (R), transaction (collect or transfer, CT) or access (CUT); the number of verbs predicted (*Pred.*), the *Total* number of verbs identified by workers, and the *Ratio* of predicted verbs to total verbs. For the remaining unpredicted verbs, we rely on worker consensus to determine the verbs' classifications.

TABLE V. INFORMATION ACTIONS FILTERED BY KNOWN ACTION LIST

Policy	Verbs matching code priors						Pred.	Total	Ratio
	C	U	T	R	CT	CUT			
Amazon	22	9	9	12	14	8	74	174	0.425
Rovio	11	23	9	8	5	5	61	149	0.409
Walmart	21	22	15	5	17	4	84	272	0.309
Waze	3	16	6	4	12	4	45	175	0.257
Zynga	11	36	25	2	15	2	95	304	0.313

Based on the predicted verbs, Table VI presents the top six workers who completed the most HITs across all five policies: the table shows a unique *Worker ID*, followed by the number of *Policies* they coded, the total number of *tasks*, the total number of *Coded Items* and *Total Expected* items from the known action list, their *Coded/Expected* ratio, number of *Ratings/HIT* and their *Avg. Grade*, which is the average worker grade across all of their coded items. Notably, the workers with the highest Coded/Expected ratios (V7, V8, and V10) have higher Ratings/HIT and relatively lower grades. These workers not only identify known verbs, but they also identify verbs that few other workers identify. Contrast this behavior with workers V4, V6 and V9, who find only half (or less) of the known verbs, yield few Ratings per HIT, and generally code items that other workers identify (i.e., they have above average grades).

TABLE VI. MOST PRODUCTIVE WORKERS EVALUATED AGAINST VERBS ON THE KNOWN ACTION LIST

Worker ID	Policies	Tasks	Coded Items	Total Expect.	Coded/ Expect.	Ratings/ HIT	Avg. Grade
V4	5	109	228	438	0.521	4.1	3.0
V6	1	98	241	425	0.567	4.2	3.2
V7	5	73	248	297	0.835	12.8	1.3
V8	3	55	186	236	0.788	10.0	1.8
V9	1	50	70	222	0.315	2.2	4.0
V10	2	35	97	134	0.724	9.8	1.8

B. Comparative Evaluation of E3 and RE'13 Case Study

We compared worker results for the Zynga policy with our RE'13 study results [7] to determine whether task decomposition produced the desired data quality for extracting privacy requirements. Figure 9 presents precision and recall for each of the four microtasks. Notably, precision and recall are inversely proportional, and precision increases with the number of raters in agreement. While all the tasks largely track the same curvature, the recall for identifying actors (i.e., sources or targets) falls sharply by comparison to actions and information types. The hourly rate for the actor identification task was also much lower by comparison. While this task may be perceivably easier given the consistent use of pronouns to describe actors, there are a significant number of such entities in the text. Thus, worker motivation may be decreasing over time which yields fewer correct answers.

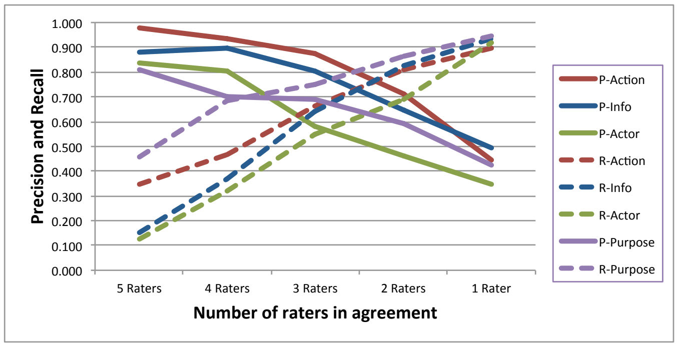


Fig. 9. Precision and Recall comparing the Crowd Worker result with our RE'13 case study result: Precision (P) and Recall (R) are inversely proportional with the number or raters who agree with the ratings.

In addition, we manually inspected the data at the 2-Rater agreement level and found that "false positives" (FP) include implied information actions and types that had previously been missed by the expert analysts in our case study, because the case study was narrowly focused on explicit statements of collection, use and transfer. For actions and 2 raters (recall = 0.811), we found 34/48 FP verbs that could have been reasonably included in our previous case study. This includes verbs, such as *provide*, *deliver*, *notify* and *record* in reference to a kind of information. Among these verbs, 11 described implied information uses and transfers, including *signing up* (for an account), *inviting* (friends) and *remembering* (your last visit). For information types (recall = 0.823), we found 99/119 FP phrases that could be reasonably included in the case study. This includes technology devices, such as mobile devices and browsers, that contain personal information. In experiment E3, however, we instructed workers to report actions performed on any information. Thus, we believe the crowdsourcing result yields a desired increase in extraction coverage.

Table VII presents the total cost for distributing the decomposed task to MTurk for the five policies: at the time of this experiment, the four microtasks cost a combined \$0.36 and were each distributed to 5 workers to yield a \$1.80 unit-cost per task for 5 raters; Amazon charges 10% in *MTurk Fees*, which is added to the *Total Cost*; the last column shows the total *Worker Compensation* that a worker receives for completing all assignments, which we believe motivates workers to participate.

Recall from Section III that the cost for two trained analysts to code the Zynga policy is \$172.50. In addition to the MTurk total cost, there is a one-time cost for sub-dividing the policies (approx. \$5). Based on our task decomposition, we reduced the manual cost by 60% for the Zynga policy and with 2-rater agreement we increased action coverage by 16% with a false positive rate of 3.6%. Based on our average cost of \$0.46 per statement coded by a single trained analyst, and a combined 522 statements for the five policies in our study, the projected cost to have two trained experts code the same five policies would be \$480.24. The task decomposition approach only cost \$305.42, which is a scaling improvement of 1.5:1. As we discuss in Section VI, we discovered additional prospects for improving this scaling factor in the future.

TABLE VII. COST TO CROWDSOURCE THE TASK DECOMPOSITION

Policy	Tasks*	MTurk Fees	Total Cost	Worker Comp.
Amazon	18	\$3.24	\$40.64	\$6.48
Rovio	18	\$3.24	\$40.64	\$6.48
Walmart	27	\$4.86	\$58.46	\$9.72
Waze	34	\$6.12	\$72.32	\$12.24
Zynga	32	\$5.76	\$68.36	\$11.52

* Assumed a \$1.80 unit-cost per task for 5 raters.

VI. Discussion and Summary

We now discuss our results from the decomposed task workflow and potential for future work. The three experiments show complimentary results: experiment E1 demonstrates the feasibility of sentence-level coding and the 5-15 worker threshold at which worker agreement stabilizes; in experiment E2, we tasked workers to apply both sentence- and phrase-level codes on a per statement-basis, wherein we discovered that the aggregate of workers' *partial responses* scored better for most sentences than what workers produced as complete responses; and in experiment E3, we leveraged this insight to create a task decomposition that uses fewer workers and simpler phrase-level coding tasks on larger portions of text to yield an acceptable aggregate response at a reduced overall cost.

Our results show that for a 1.5:1 cost reduction, we can use crowdsourcing to scale requirements extraction for privacy policies. This ability to scale could enable more efficient data flow research in larger data ecosystems [7], as well as comparative studies that contrast practices across business types (e.g., advertisers, social networking sites, etc.) In addition, while trained analysts may fatigue due to the rote task of coding large policies, the task decomposition approach may avoid this fatigue by leveraging a larger population of coders over much smaller tasks (2-3 sentences at a time).

We envision augmenting our crowdsourcing approach with advanced NLP-based techniques that leverage the worker-supplied data. This includes dependency parsers that can help identify actors in the source and target microtask, as well as relevant verbs covering worker-supplied information types. To improve these automated techniques, we plan to build an information type ontology based on these results after conducting additional worker processing to identify synonyms and subsumption relationships. Finally, for machine-learning based NLP, we believe our crowdsourced approach may produce the large training sets needed to train classifiers.

The results show that requirements analysts and researchers can scale requirements extraction using task decomposition and untrained crowd workers, wherein a worker is *untrained* if they are untested against a training regime prior to performing the task. With appropriate pilot studies, we discovered the minimum number of workers, work quality metrics, task size and payment amount for this kind of work (coding text). We envision that other requirements acquisition or analysis tasks will require similar pilot testing to arrive at respective task-specific numbers and can benefit from our approach.

Acknowledgment

We thank Hanan Hibshi, Sudarshan Wadkar, and Anastasia Timoshenko for their helpful feedback. This work was supported by NSF Award #1330596.

References

[1] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum. "reCAPTCHA: Human-Based Character Recognition via Web Security Measures." Science 321 (5895): 1465–68, 2008.

[2] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H.R. Motahari-Nezhad, E. Bertino, S. Dustdar. "Quality Control in Crowdsourcing Systems: Issues and Directions." *IEEE Int. Comp.* 17 (2): 76–81, 2013.

[3]E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2009.

[4] V. Ambati, S. Vogel, J. Carbonell. "Collaborative Workflow for Crowdsourcing Translation." *ACM Conf. Comp. Sup. Coop. Work*, pp. 1191–94, 2012.

[5] T.D. Breaux, *Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems*. Ph.D. Thesis, N.C. State Univ., Apr. 2009

[6] T.D. Breaux, A.I. Antón, J. Doyle. "Semantic Parameterization: A Process for Modeling Domain Descriptions." *ACM Trans. Soft. Engr. Method.*, 18(2): 5, Nov. 2008

[7] T.D. Breaux, H. Hibshi, A. Rao. "Eddy, A Formal Language for Specifying and Analyzing Data Flow Specifications for Conflicting Privacy Requirements." To Appear: *Req'ts Engr. J.*, 2014.

[8] T.D. Breaux, M.W. Vail, A.I. Antón: "Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations", *IEEE Int'l Req'ts Engr. Conf.*, pp. 46-55, 2006.

[9] P. Caire, N. Genon, P. Heymans, D.L. Moody. "Visual notation design 2.0: Towards user comprehensible requirements engineering notations," *IEEE Int'l Req'ts Engr. Conf.*, pp. 115-224, 2013.

[10] L. B. Chilton, G. Little, D. Edge, D. S. Weld, J. A. Landay. "Cascade: Crowdsourcing Taxonomy Creation." *Conf. Human Factors in Comp. Sys.*, pp. 1999–2008, 2013, ACM.

[11] J. Cleland-Huang, A. Czauderna, M. Gibiec, J. Emenecker "A machine learning approach for tracing regulatory codes to product specific requirements," *IEEE Int'l Conf. Soft. Engr.*, pp. 155-164, 2010.

[12] J. Corbin, A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, SAGE Pubs., 2007.

[13] A. Doan, R. Ramakrishnan, A. Y. Halevy. "Crowdsourcing Systems on the World-Wide Web." *Commun. ACM* 54 (4): 86–96, 2011.

[14] C. Eickhoff, A. P. de Vries. "Increasing Cheat Robustness of Crowdsourcing Tasks." *Information Retrieval* 16 (2): 121–37, 2013.

[15] A. Finnerty, P. Kucherbaev, S. Tranquillini, G. Convertino. "Keep It Simple: Reward and Task Design in Crowdsourcing." *Biannual Conf. Italian Chapter of SIGCHI*, 2014, ACM.

[16] D.G. Gordon, T.D. Breaux, "Assessing regulatory change through legal requirements coverage modeling." *IEEE Int'l Req'ts Engr. Conf.*, pp. 145-154, 2013.

[17] J. J. Horton, L. B. Chilton. "The Labor Economics of Paid Crowdsourcing."*ACM Conf. Electronic Commerce*, pp. 209–18, 2010.

[18] P.-Y. Hsueh, P. Melville, V. Sindhwani. "Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria." *NAACL HLT 2009 W'shp Active Learning for NLP*, pp. 27–35. 2009.

[19] N. Q. V. Hung, N. T. Tam, L. N. Tran, K. Aberer. "An Evaluation of Aggregation Techniques in Crowdsourcing." *Web Info. Sys. Engr.*, 2013, Springer Berlin Heidelberg.

[20] P. G. Ipeirotis, F. Provost, J. Wang. "Quality Management on Amazon Mechanical Turk." *SIGKDD Workshop on Human Computation*, pp. 64–67. 2010, ACM.

[21] E. Kamar, S. Hacker, E. Horvitz. "Combining Human and Machine Intelligence in Large-Scale Crowdsourcing." *Conf. Auto. Agents & Multiagent Sys.*, pp. 467–74, 2012.

[22] G. Kaur, "Analyzing Email Archives to Better Understand Legal Requirements," *W'shp Req'ts Engr. & Law*, 2009, pp. 21-26.

[23] G. Kazai. "An Exploration of the Influence That Task Parameters Have on the Performance of Crowds." *CrowdConf* 2010.

[24] A. Kittur, E. H. Chi, B. Suh. "Crowdsourcing User Studies with Mechanical Turk." *SIGCHI Conf. Human Factors in Comp. Sys.*, pp. 453–56, 2008, ACM.

[25] A. Kittur, B. Smus, S. Khamkar, R. E. Kraut. "CrowdForge: Crowdsourcing Complex Work." *ACM Symp. User Interface Software and Technology*, pp. 43–52, 2011, ACM.

[26] A. Kulkarni, M. Can, B. Hartmann. "Collaboratively Crowdsourcing Workflows with Turkomatic." *Conf. Comp. Sup. Coop. Work*, pp. 1003–12, 2012, ACM.

[27] S. L. Lim, A. Finkelstein. "StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation," *IEEE Trans. Soft. Engr.*, 38(3): 707-735, 2012.

[28] G. Little, L. B. Chilton, M. Goldman, R. C. Miller. "TurKit: Human Computation Algorithms on Mechanical Turk." ACM Symp. UI Soft. & Tech., pp. 57–66, 2010.

[29] A.K. Massey, J. Eisenstein, A. Anton. "Automated text mining for requirements analysis of policy documents." *IEEE Int'l Req'ts Engr. Conf.*, pp. 4-13, 2013.

[30] S. Messick, D.N. Jackson. "Acquiescence and the factorial interpretation of the MMPI." *Psych. B.*, 58(4): 299-304, 1961.

[31] M. Negri, L. Bentivogli, Y. Mehdad, D. Giampiccolo, A. Marchetti. "Divide and Conquer: Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora." Conf. Emp. Meths. in NLP, pp. 670–79, 2011, ACL.

[32] G. Paolacci, J. Chandler, P. G. Ipeirotis. "Running Experiments on Amazon Mechanical Turk." *Judge. & Dec. Making* 5 (5): 411–19, 2010.

[33] A. J. Quinn, B. B. Bederson. "Human Computation: A Survey and Taxonomy of a Growing Field." *Conf. Human Factors in Comp. Sys.*, pp. 1403–12, 2011, ACM.

[34] K. Ryan, "The Role of Natural Language in Requirements Engineering," *Int'l Symp. Req'ts Engr.*, 1993, pp. 240-242.

[35] J. Saldana, *The Coding Manual for Qualitative Researchers*, SAGE Pubs, 2012.

[36] R. Snow, B. O'Connor, D. Jurafsky, A. Y. Ng.. "Cheap and Fast—but Is It Good?: Evaluating Non-Expert Annotations for Natural Language Tasks." *Conf. Emp. Meths. in NLP*, pp. 254–63, 2008, ACL.

[37] A. Sheshadri, M. Lease. "SQUARE: A Benchmark for Research on Computing Crowd Consensus." *AAAI Conf. Human Comp. & Crowdsourcing*, 2013, AAAI.

[38] O. Zaidan, C. Callison-Burch. "Crowdsourcing Translation: Professional Quality from Non-Professionals."ACL, pp. 1220–29, 2011.

Appendix A.

A. Statement-level Coding Frame

The statement-level coding frame was established in a case study to extract data flow requirements from privacy policies [7]. In this study, relevant statements in a policy were classified as one of the following (irrelevant statements were not coded):

Collect –	any act by a first party to access, collect, obtain, receive or acquire data from another party;
Consent –	any act by a party to consent to, or control the use of, their personal information;
Use –	any act by a first party to use data in any way for their own purpose;
Retain –	any act by a first part to retain data for a particular period of time or in a particular location; and
Transfer –	any act by a first party to transfer, move, send or relocate data to another party.

B. Phrase-level Coding Frame

After the statement-level code is assigned, the analyst would then apply a phrase-level coding frame corresponding to semantic roles [6]. These roles consist of the following codes:

Modality –	whether the action is a permission, obligation or prohibition;
Subject –	the actor who performs the action on the datum;
Datum –	the information on which the action is performed;
Purpose –	the purpose for which the action is performed; this role applies to any action, but is especially necessary to describe use of the datum.
Source –	the source from which information is collected;
Target –	for transfer actions, the recipient to whom the information is transferred.