Skierarchy: Extending the Power of Crowdsourcing Using a Hierarchy of Domain Experts, Crowd and Machine Learning

TREC 2012 Crowdsourcing Track Paper
Ramesh Nallapati[⌘], Sanga Peerreddy[§] and Prateek Singhal[§]
{nmramesh,sanga,prateek.singhal} @setuserv.com
SetuServ, Inc.
[⌘]330 Elan Village Lane, #223, San Jose, CA, USA 95134
[§]Orange 108, My Home Rainbow, Shekpet, Hyderabad, AP India 500008

1 Abstract

In the last few years, crowdsourcing has emerged as an effective solution for large-scale 'micro-tasks'. Usually, the micro-tasks that are accomplished using crowdsourcing tend to be those that computers cannot solve very effectively, but are fairly trivial for humans with no specialized training. In this work, we aim to extend the capability of crowdsourcing to tasks that are complex even from a human perspective.

Towards this objective, we present a novel hierarchical approach involving a small number of domain experts at the top of the hierarchy, a large crowd with generic skills at the intermediate level, and a Machine Learning system serving as a personal assistant to the crowd, at the bottom level. We call this approach *Skierarchy*, short for Hierarchy of Skills.

To test the efficacy of the Skierarchy approach, we deployed the model on the TREC 2012 TRAT task, a task we believe is fairly complex compared to typical micro-tasks. In this paper, we present illustrative experiments to demonstrate the utility of each of the layers of our hierarchy. Our experiments on TRAT as well as IRAT show that using an interactive process between the experts and the crowd could significantly reduce the need for redundancy among the crowd, while also enabling a crowd with generic skills to perform tasks that are reserved for specialists. Further, we found from our TRAT experience that both the crowd and the Machine Learning system improve their performance over time as they gain experience on specialized tasks.

2 Motivation

Crowdsourcing has recently emerged as a scalable and cost-effective technological alternative to performing tasks that are either too prohibitively expensive to be performed by expert humans or too complex for machines. In this process, a large-scale project is broken down into several repeatable, independent "micro-tasks" that are then posted on the Internet for users across the world to solve on a pay-per-task basis. This process has been highly successful on many tasks such as digitization of books (von Ahn L. M., 2008), adult content moderation[1], etc.

With the rising success of crowdsourcing, considerable research effort has been invested on improving the workflow processes in order to optimize quality and cost effectiveness of the output. Several novel techniques have been developed to address this issue, such as pre-filtering the crowd by subjecting them to qualification tests, using redundancy to improve accuracy, inserting data with known ground truth to filter out spammers, etc. (Aniket Kittur, 2008).

One question about crowdsourcing that is still not very well researched upon is the degree of complexity of tasks that a crowdsourcing-based solution is capable of solving effectively. Most of the success stories of crowdsourcing are tasks that are hard for a computer to solve, but fairly trivial for an untrained human (e.g.: image annotation tasks, key-word search tasks, finding business locations etc. that are typically found on Amazon Mechanical Turk). It is not clear if crowdsourcing can be extended towards solving more complex problems that data analysts encounter in highly specialized domains such as legal, medical and educational informatics. Some research is just beginning to emerge on breaking complex tasks in specific applications into simpler micro-tasks that are suitable for crowdsourcing (Bernstein, 2010).

[1] CrowdFlower.com has a real-time app that performs image moderation using crowdsourcing at http://crowdflower.com/rtfm/

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Skierarchy: Extending the Power of Crowdsourcing Using a Hierarchy of Domain Experts, Crowd and Machine Learning	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SetuServ, Inc,330 Elan Village Lane, #223,San Jose,CA,95134**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Twenty-First Text REtrieval Conference (TREC 2012) held in Gaithersburg, Maryland, November 6-9, 2012. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA). U.S. Government or Federal Rights License

14. ABSTRACT

In the last few years, crowdsourcing has emerged as an effective solution for large-scale ?micro-tasks?. Usually, the micro-tasks that are accomplished using crowdsourcing tend to be those that computers cannot solve very effectively, but are fairly trivial for humans with no specialized training. In this work, we aim to extend the capability of crowdsourcing to tasks that are complex even from a human perspective. Towards this objective, we present a novel hierarchical approach involving a small number of domain experts at the top of the hierarchy, a large crowd with generic skills at the intermediate level, and a Machine Learning system serving as a personal assistant to the crowd, at the bottom level. We call this approach Skierarchy, short for Hierarchy of Skills. To test the efficacy of the Skierarchy approach, we deployed the model on the TREC 2012 TRAT task, a task we believe is fairly complex compared to typical micro-tasks. In this paper, we present illustrative experiments to demonstrate the utility of each of the layers of our hierarchy. Our experiments on TRAT as well as IRAT show that using an interactive process between the experts and the crowd could significantly reduce the need for redundancy among the crowd, while also enabling a crowd with generic skills to perform tasks that are reserved for specialists. Further, we found from our TRAT experience that both the crowd and the Machine Learning system improve their performance over time as they gain experience on specialized tasks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

In this work, we are primarily interested in exploring the feasibility of extending crowdsourcing-like solutions to sophisticated data analytics problems in specialized domains. Our main contribution is a novel hierarchical approach called "Skierarchy" that utilizes highly specialized domain experts at the top of the hierarchy, who train and supervise the crowd with generic skills sitting at the intermediate level, and a Machine Learning system at the bottom of the hierarchy acting as a personal assistant to the crowd. Although using experts in combination with people with generic skills is a successful technique in other domains (e.g.: Doctors working with Nurse Practitioners in the medical domain), it has not been sufficiently experimented with in the crowdsourcing setting. Also, although some work on combining crowdsourcing with Machine Learning is beginning to appear (Alexander J. Quinn, 2010), we believe ours is the first effort that proposes using Machine Learning as personal assistant to the crowd.

The Text Relevance Assessing Task (TRAT) is an example of a moderately complex task that is typically performed by domain experts[2], and therefore it forms an ideal testing ground to validate our Skierarchy approach. In the rest of the paper, we first describe the Skierarchy approach, and its application to TRAT and IRAT. Then, we present our internal experiments to demonstrate the effectiveness of various layers in the hierarchy. We then report performance numbers of our system on the official evaluation.

3 Skierarchy Approach

The Skierarchy approach aims to extend the capabilities of crowdsourcing to solving sophisticated domain specific data analytics tasks that are currently performed by domain experts. Our approach retains the best practices of crowdsourcing technologies but adds additional layers in the workflow in order to accomplish this goal, which we describe below.

3.1 Training and Supervision of the crowds using Domain Experts

Although crowdsourcing is very cost effective for many data annotation tasks, it has not yet been able to replace highly expensive domain experts in many verticals such as legal, medical and educational informatics. For example, recommendation systems such as Pandora[3], Art.sy[4], Lex Machina[5] and eSparkLearning[6] employ a large number of domain experts to curate their respective domain-specific data. We believe the main hindrance to crowdsourcing these tasks is the relative inability of the crowd to understand and interpret domain specific language/data. Clearly, domain experts are indispensible for these tasks. However, we can use them most effectively by moving them to a supervisory role, to design the micro-tasks, and train and support the less skilled crowd. Following this idea, we propose a hierarchical solution where we use a small number of domain experts to train and supervise a large crowd, so that we still retain the ability to scale at low costs.

The responsibility of domain experts includes (a) breaking down the tasks into generalizable micro-tasks, so that the tasks become relatively "de-skilled", and therefore more amenable to crowdsourcing, (b) training and supervising the crowd actively in performing these micro-tasks, and (c) solving a small number of micro-tasks themselves, that the crowd finds too difficult or confusing.

This approach enables us not only to extend the capability of crowdsourcing to more complex domain-specific tasks by letting the domain experts focus on the difficult problems, and shifting the simpler ones to the crowd, but also in improving the quality of the crowd's output over time through training and supervision. Over time, we expect that members of the crowd become experts at specific tasks that are domain independent.

---

2 In fact, the ground truth "qrels" for TREC topics were curated by full-time information analysts at LDC. Besides, TREC topics tend to be very nuanced with their inclusion and exclusion rules compared to broader subject categories such as politics, sports or entertainment.

3 www.pandora.com is a music recommendation engine.

4 www.art.sy recommends works of art to art-lovers.

5 www.lexmachina.com creates a structured knowledge base from unstructured legal dockets.

6 www.esparklearning.com recommends learning aids such as apps and videos to students based on their learning needs.

3.2 Assisting the crowd using an Automatic Machine Learning system

The performance of the crowd is expected to improve under the supervision of domain experts. However, since the domain experts do not micro-manage the crowd on every micro-task, we use Machine Learning as an additional layer in the hierarchy to improve the performance of the crowd. Unlike traditional settings where Machine Learning either replaces humans wholly or partially, we use Machine Learning as a personal assistant to the crowd rather than as an alternative. Our main idea is based on the observation that modern Machine Learning models, although not on par with humans in terms of performance on most tasks, are still rich enough to assist humans by:

(a) Emitting prediction scores for each micro-task, which can be used as a surrogate measure for its difficulty. The difficulty measure may in turn be used to (i) allocate larger chunk of time to the more difficult items, (ii) assign the more difficult items to the better performers, and (iii) measure the throughput of each member in the crowd normalized by difficulty of the micro-tasks he/she has solved.

(b) Offering annotation suggestions to the annotator to reduce his/her cognitive load,

(c) Directing the annotator's attention by highlighting relevant sections in the data, and

(d) Detecting potential human errors through their prediction mechanism and alerting the crowd.

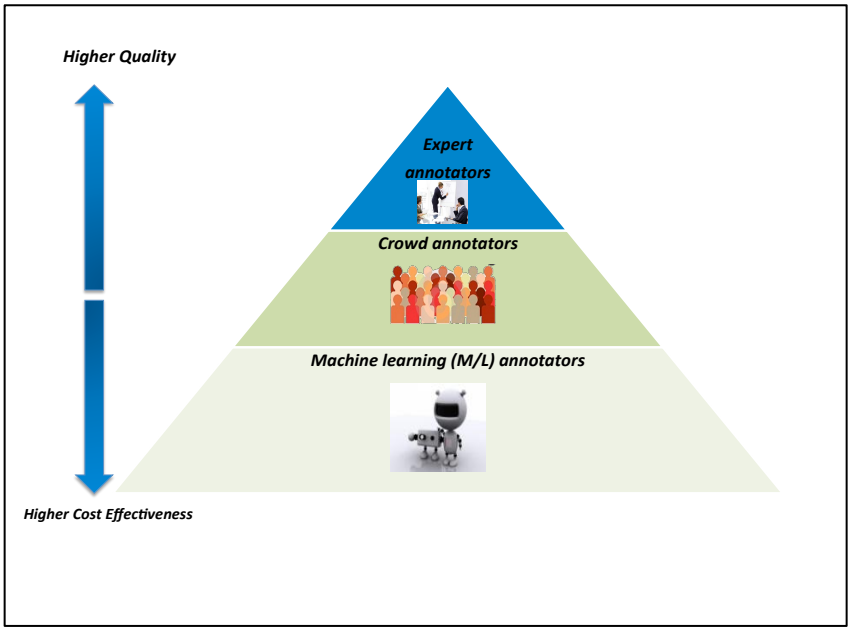


Figure 1: The Skierarchy model: Hierarchy of Skills

The schematic diagram in Figure 1 presents a unified view of the Skierarchy model with its three layers comprising domain experts, crowd and machines. As the figure indicates, the top of the pyramid consists of the domain experts who offer the highest quality but the least cost-effectiveness and scale, while the base comprises Machine Learning that offers lowest quality but the highest scale and cost effectiveness. The crowd is in the intermediate layer offering mid-level quality, costs and scale.

3.3 "Open-Crowd" vs. "Closed Crowd"

The gold standard of crowdsourcing industry is to employ what we call "open-crowd", a set of anonymous people from around the world who perform the micro-tasks on a pay-per-task basis, through a web-based platform. This has been a very attractive setting since open-crowd is very elastic (i.e., the size of the crowd can be increased or decreased at a moment's notice), scalable, cheap and available round the clock.

In contrast to the standard setting, we used the setting of what we refer to as "closed-crowd" in our experiments, where we hire employees in India with generic college background and good reading comprehension skills in English, and bring to them to the office space to work for us full-time on micro-tasks. This setting was necessitated for our initial experiments because our Skierarchy approach requires close interaction between the domain experts and the crowd, although we believe it could be extended in the future to the open-crowd setting as well. Although closed-crowd setting is less elastic, we found several advantages of this setting in our experiments, some of which we list below:

Fast prototype cycle: Closed crowd is ideal for experimenting and developing a prototype process - it makes the feedback loops short and quick. This is not different from the setting of a manufacturing firm that outsources its mature processes to third parties but performs its own manufacturing during the prototyping stage. Also, it is easier to collect plenty of data about the right way to design a micro-task by observing the crowd in the initial stages.

Training, Knowledge sharing and Continuity: We found face-to-face training and feedback to be very effective in bringing the crowd up-to-speed on a complex micro-task. In addition, we allowed the crowd to share tips based on their experience with their peers during the training phase, which they found very useful in improving their skills. The other benefit of training and employing the same crowd for a long period is that the crowd improves over time, and provides higher quality as time progresses.

Privacy concerns: Although many companies have a pressing need to analyze terabytes of internal data that they generate through their transaction records, customer support logs, etc., they are reluctant to employ open-crowd systems due to the privacy and security concerns inherent in the data. We believe a closed-crowd setting alleviates these concerns to a great extent.

4 Text Relevance Assessment Task

4.1 Experimental setting and tools

We employed 5 people with college degrees in India as our "closed-crowd"[7], and used 1 person with good experience in information analytics, as our domain expert[8]. The crowd was screened at the time of hiring to make sure that they had reasonable reading comprehension skills in English. The expert spent only 1/3rd the time as each member of the crowd in the entire annotation process.

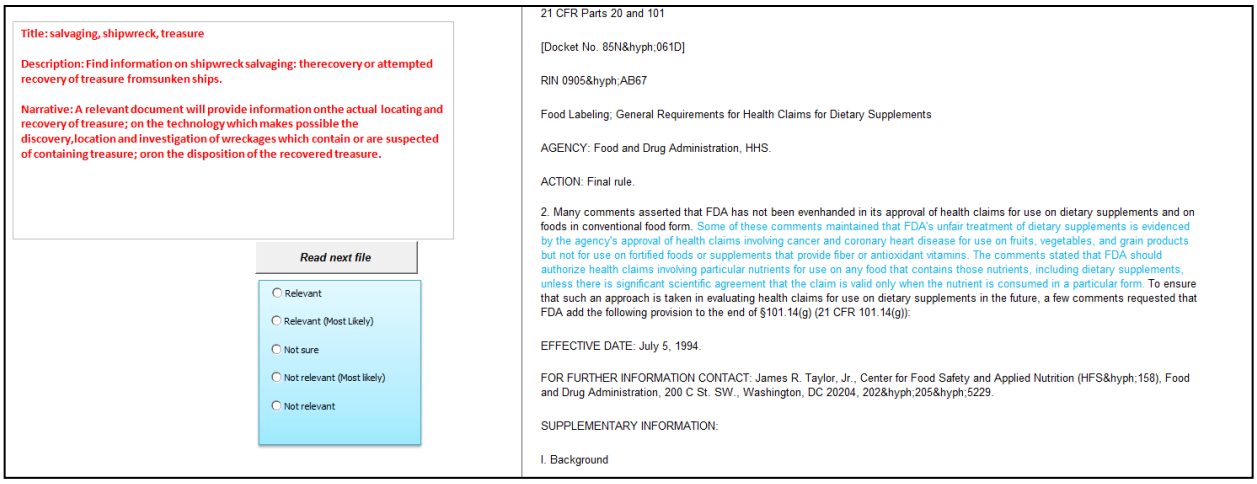


Figure 2: TRAT UI Screenshot

We used the Logistic Classifier from Stanford CoreNLP[9] as our Machine Learning system. We wrote our own parsers to parse the SGML format of the TREC corpus to extract the subset of 18,600 documents needed for our experiments. For UI, we wrote scripts in *Excel* using VBA, to support user highlighting as well as Machine highlighting. The UI, a snapshot of which is shown in Figure 2, also has radio buttons to enter ratings, which get automatically recorded into a spreadsheet. Additionally, we inserted items with known ground truth at random, and built real-time alerts as part of our UI in cases where the crowd's annotations deviated from the ground truth. The documents are displayed one at a time to the user along with the topic description on the side. The time a user spent on each document is recorded as the time between two clicks of the "Next Document" button.

We used annotations on a scale of 1-5 where 5 represents highest relevance and 1 for highest non-relevance. A rating of 3 was assumed to mean that the annotator considered the document as either too confusing or too difficult to judge for relevance.

---

7 See Section 3.3 above for the meaning of "closed-crowd".

8 Please note that our original plan was to hire a high school English teacher but we hired an information analyst instead due to time constraints. We believe that an English teacher would have done an equally good job as a domain expert.

9 http://nlp.stanford.edu/software/corenlp.shtml

As we will describe below in more detail, we used an iterative approach to annotations as opposed to a parallel approach (Greg Little, 2010). We used 2 iterations to perform the annotations, which in-effect means each document received at-most two annotations (and at-least one annotation).

4.2 Description of the Skierarchy Workflow process

Step 1: Training the Crowd Using the Expert: For each topic, we first used the averaged ranking of TREC8 submission runs as our starting point. The crowd was first asked to annotate the top 20 documents in the ranked list independently, and also highlight passages of text that they deem relevant to the topic. The expert then examined the documents that had disagreements between them and assigned a final rating based on his own judgment. A meeting was held with the crowd in which the expert walked over each document in the disagreement set, examined their passage highlights to understand their reasoning, and explained to them the reasons behind his own judgment. This was considered as an initial training period for the crowd. Upon the completion of the initial training period, the crowd completed annotations for ~300 documents. These documents were checked randomly by the expert to ensure high quality.

Step 2: Building a Machine Learning Model, and a Curated Keyword Set: The annotation data was then used to train the Machine Learning algorithm, which in this case is a Logistic Regression based binary classifier. For the purposes of training the binary classifiers, we mapped the ratings of 4 and 5 to positive class and the rest to the negative class. The logistic classifier also trains on the passages highlighted by the crowd, treating them simply as additional documents. We also use the 20 words with the highest weights for the positive class, as learned by the classifier and filter them manually using the expert, to generate a curated list of keywords for each topic. The Machine Learning model is then run on the remaining documents on that topic to generate prediction scores in the range [0,1], where a score of `1' indicates that the machine thinks that the document is absolutely relevant to the topic, and a score of `0' indicates the machine's belief that the document is absolutely non-relevant to the topic.

Step 3: Annotations with the Help of both Machine Learning and the Expert: The documents are then divided into several buckets by binning them based on the Machine Learning scores, and each bucket is provided to a distinct annotator. Typically, the annotators with superior performance (judged during the training period for the topic) were given the buckets with higher scores and the annotators with moderate performance were given the buckets with the lowest scores. The rationale is that the documents with higher scores tend to contain larger proportion of relevant documents, so the annotator needs to be careful to avoid misses, while the ones with lower scores tend to be mostly non-relevant, so a moderate performer will not hurt too much. When the documents are displayed to the annotators, we also highlight any words that match the list of curated keywords for that topic. In this case, the expert annotator only examines the documents assigned a rating of 3 and assigns a final rating. The expert also occasionally inspects a few documents at random to ensure that the crowd is on track.

Step 4: Automatic Error Correction with Machine Learning: Once the entire set of documents is annotated, we retrain the Machine Learning model using the complete annotated data and generate predictions for all the documents using 10 fold cross-validation. Next, we do a second pass of annotations with the crowd focusing on documents where the Machine Learning scores and annotations are in disagreement. For example, the crowd re-examines the documents with scores 1 and 2 (likely non-relevant) sorted in the descending order by the Machine Learning scores, and documents rated 4 and 5 (likely relevant) sorted in the ascending order of Machine Learning scores. These are the documents that may contain likely annotation errors. We made sure that each documents in this round is assigned to annotator that is different from its original annotator. Also the new annotator has no access to the original annotation to prevent potential biases. We also generate a new set of curated keywords based on the updated ML model and use them for keyword highlighting in this stage.

4.3 Validation Experiments

In this section, we aim to test whether some of the hypotheses we made in the Skierarchy approach hold true in the TRAT task. Note that we performed these experiments after the official submission since we were not allowed to access TREC ground truth before the submission. We present each hypothesis in the form of a question, and then proceed to answer empirically through our internal experiments. We have presented our results with respect to self-curated judgments to account for ambiguity in the topic definitions in TRAT.

Does machine assistance boost productivity and quality of the crowd?

In order to quantify the improvement in productivity and quality as a result of using Machine Learning output, we performed three independent rounds of annotations with the following ML assistance strategies:

- *Top 10 keywords*: ML assists by highlighting top 10 keywords learned from training data in a given document,
- *Curated keywords*: ML suggested keywords are curated by the expert for each topic, and are highlighted in a given document,
- *Curated keywords and prioritization*: In addition to curation of keywords and highlighting them, the documents are presented to the crowd in the decreasing order of ML confidence score. Crowd was instructed to expect a higher percentage of relevant documents at the top than at the bottom of the list.

Above three ML assistance strategies were deployed on 3 sets containing 35 documents each to create 9 simulations. Documents in each set were chosen at random so that the mean and range of ML confidence scores were the same across the 3 sets. Each simulation (i.e. set and ML assistance combination) was routed through 3 annotators according to the following table so that we could control for the crowd bias and productivity levels to some extent.

ML Assistance Strategy	Set A	Set B	Set C
Top 10 keywords	Ann1	Ann2	Ann3
Curated keywords	Ann2	Ann3	Ann1
Curated keywords & prioritization	Ann3	Ann1	Ann2

Crowd productivity and quality were measured in each simulation in terms of time per annotation and accuracy respectively. The measurements showed that the crowd productivity increased by 11% when curated key words were highlighted instead of the top 10 keywords. Productivity further increased by 36% when the documents were prioritized by the ML confidence score. In this experiment, gain in quality was marginal (~2%) as the crowd already had high accuracy levels (95%) leaving little room for improvement with keyword-curation or prioritization. In settings when such high accuracy levels are not possible (e.g., when no training or coaching are provided), keyword-curation and prioritization may be able to help with quality as well.

Does Expert in the loop improve quality and productivity?
Expert can add value to the crowd through a) training/coaching, which improves their capability, and b) by completing annotations themselves where the crowd is likely to commit mistakes. Current crowdsourcing techniques focus on achieving the latter through redundancy across the crowd. Here, we aim to compare redundancy with the expert to redundancy across the crowd.

To test this, we picked a set of 50 random documents for a fairly complex topic. Four annotators and the expert annotated these documents independently on a scale of 1 to 5. Quality is evaluated against TREC relevance judgments, some of which we corrected to comply with the topic descriptions[10]. Following table summarizes the results of our analysis:

Number of annotations by category and annotator

	Ann1	Ann2	Ann3	Ann4	Majority	Expert	TREC
Confusing (rating of 3)	5	7	8	4	7	0	0
False Positive	1	0	1	2	0	0	1
False Negative	2	2	2	1	2	0	3

As can be seen from above table, majority opinion did not help clarify the confusing annotations. This was partly because of ambiguity in the topic description and narrative and high level of comprehension needed to understand some articles. In such cases, most annotators tend to err simultaneously, rendering majority voting ineffective. However, having expert in the loop resulted in resolution of all these errors. Also, the expert was able to annotate tasks within half as much time as the crowd took. This efficiency gain partly offsets the higher cost incurred in hiring experts. In this experiment, we also noticed that the crowd is fairly good at telling when they are not confident. The expert's time can be best spent in such situations. In our TREC annotations, approximately 70% of the expert time was spent in analyzing documents that were confusing to the crowd or documents where the crowd strongly disagreed with ML-based suggestions.

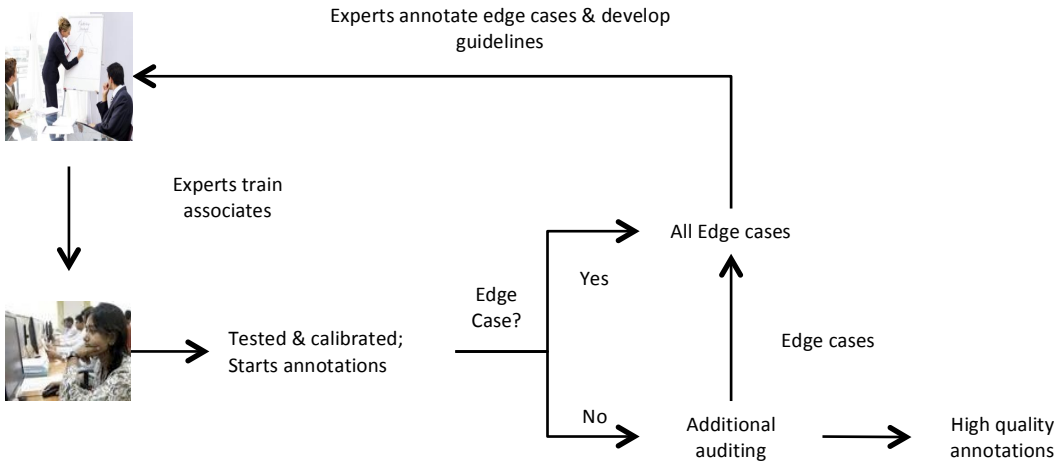
[10] In the 50 documents we chose, 4 documents were corrected for ground truth (1 false negative and 3 false positives).

What type of performance management can improve quality in crowdsourcing?

Quality is hard to measure and manage with annotations for two reasons - a) ambiguity in topics & documents leaves significant room for interpretation and subjectivity b) expected outputs are not known upfront, making quality measurable only post-hoc. Accounting for these factors in the performance management process is the key to deliver high quality annotations.

We developed a performance management process where we ask the crowd to flag the ambiguous tasks, and we hold them accountable only for the quality of the annotations that they were confident about (i.e. non-ambiguous tasks). Further, we also asked each worker to provide a short description that justifies their annotations, and highlight the supporting text in the document. We then asked a second worker to read the description and highlighted text from the first worker, and confirm the annotation. Annotations from the first and second workers were audited by an expert at random. The documents that were marked as ambiguous by the crowd are also routed to expert. This process is illustrated below –

Annotation Process Flow



To test how well this process works, we implemented this process on a set of 165 documents that were confusing to the machine. Quality is evaluated against TREC relevance judgments. Following table summarizes the results of our analysis:

Number of annotations by category and annotator

	Annotations after the 1st annotator	Annotations after the auditing from 2nd annotator
False Positives	2	1
False Negatives	5	2
Edge cases identified	15	18
Total annotations	165	165

As can be seen from above table, this process resulted in highly accurate annotations with an overall redundancy of just 1.5 (the 2nd annotator was able to do tasks twice faster using the supporting text provided by the 1st annotator although it could have led to some bias). This process also helped us develop 18 "edge" cases. Expert annotated these cases, and used them to develop additional guidelines. This in turn helped the crowd get better over time.

4.4 Our Results on TRAT

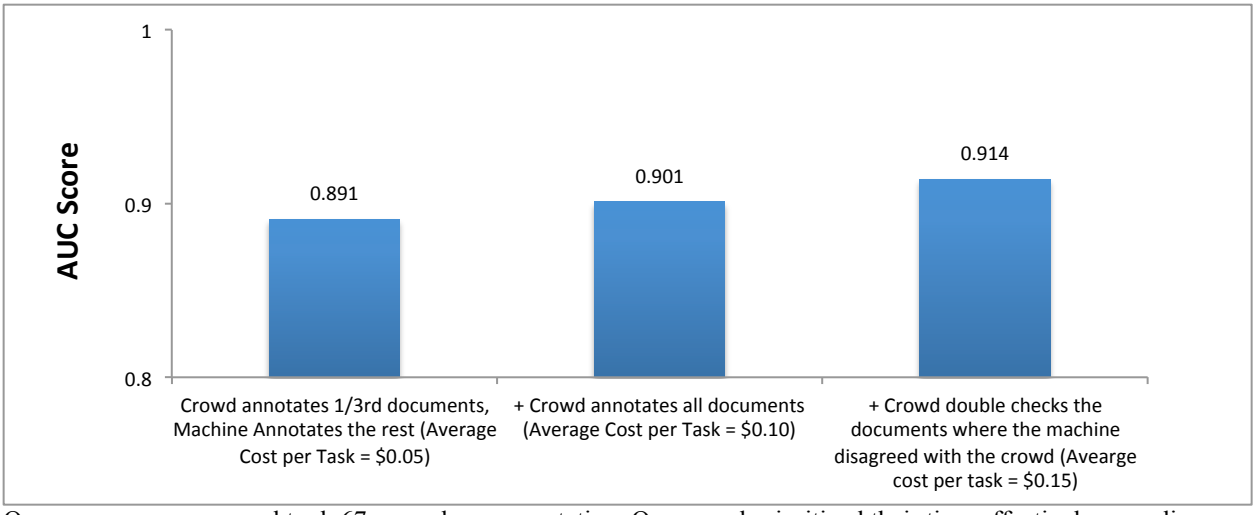
Our relevance annotations were found to be highly accurate as evaluated with respect to those of adjudicated annotations. Our performance was on par with that of of NIST's own professional assessors (TREC 8 QRels).

Run	LAM	AUC	MAP RMSE	MAP Tau
SetuServ	0.07	0.91	0.02	0.93
NIST (TREC 8 QRels)	N/A	N/A	0.03	0.96
UIowaS02r	0.05	N/A	0.08	0.77
INFLB2012	0.13	N/A	0.09	0.72
NEUEIo3	0.18	0.75	0.11	0.64
yorku12cs03	0.22	0.48	0.11	0.64
BUPTPRISZHS	0.23	N/A	0.18	0.64
OrcVBW16Conf	0.26	0.81	0.20	0.35

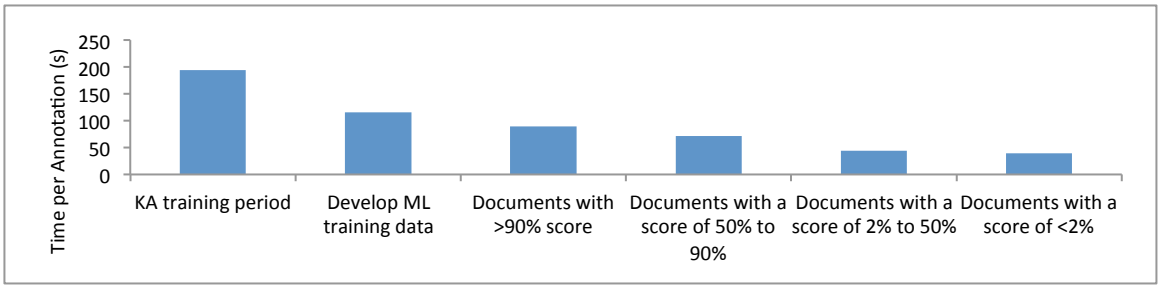
Our quality metrics were higher than the median scores across all topics as shown in the following table. (Our LAM score was 62.5% lower than the median and our AUC score was 17% higher than the median)

	Topic	411	416	417	420	427	432	438	445	446	447	Average
LAM	Median	0.15	0.16	0.2	0.17	0.18	0.27	0.26	0.19	0.21	0.08	0.19
	SetuServ	0.078	0.027	0.082	0.069	0.038	0.163	0.069	0.052	0.116	0.007	**0.07**
AUC	Median	0.86	0.85	0.75	0.71	0.73	0.71	0.78	0.83	0.82	0.76	0.780
	SetuServ	0.968	0.965	0.89	0.96	0.956	0.712	0.951	0.935	0.834	0.968	**0.914**

As shown in the following graph, comparison of the quality metrics across the first and last bars show that we achieved high levels of quality and saved quite a bit of crowd effort if we had used ML annotations for documents with ML score of <2% (i.e. where machine was confident about non-relevance of the document).



On an average, our crowd took 67 seconds per annotation. Our crowd prioritized their time effectively, spending more time when quality was required (e.g. during training period and while developing training data for M), as shown in the following table:



In our first pass, we had only one person from the crowd annotate each document. We then identified likely errors using ML confidence score, and had a second person from the crowd annotate the documents again. This resulted in an overall redundancy of 1.5. Additionally, the expert annotated 16% of documents that were confusing to the crowd, taking the redundancy to 1.66.

5 Image Relevance assessment task

5.1 Experimental Setup

For IRAT, we used the same crowd that we used for TRAT, but significantly enhanced the workflow and UI to adapt to the requirements of IRAT. In the UI, we displayed all the images for a topic together, saving sequential traversing time which otherwise would have been comparable to the assessment time for the Image task. As shown in Figure 3, the UI has radio buttons for each image to enter ratings, which get automatically recorded into a spreadsheet along with the time when the user clicked the radio button. Although ML was not used to assist crowd, we created rule-based "double-check" alerts in the UI using the textual information listed in the caption. These alerts helped retain the due diligence from the crowd.

Given that the low level of ambiguity in IRAT, we used annotations on a less granular scale of 1-3, where 3 represents highest relevance and 1 for highest non-relevance. A rating of 2 was assumed to mean that the annotator considered the document as either too confusing or too difficult to judge for relevance.

We used a combination of parallel and iterative approaches to do annotations as opposed to using only one of them (Greg Little, 2010). Our expert's direct interaction with the crowd enabled us to conduct quick experiments to deduce the optimal mix of iterative and parallel approaches, by evaluating the quality and cost tradeoffs. We used 2 annotators to perform annotations in parallel, and another annotator to annotate the images where the first two

Figure 3: IRAT UI Screenshot

disagreed, which in-effect meant that each document received at-most three annotations (and at-least two annotations). Further the expert also spent 5% of the time of a single annotator in training, supervising and annotating the confusing images.

5.2 Description of the Skierarchy Process

Step1: Training the Crowd using the Expert: The expert trained the crowd on 5 topics and provided a list of web based resources that they can refer to (e.g. how to optimally use Google image search and Wikipedia resources). The crowd was also trained to use the UI designed for IRAT effectively, e.g., focusing on reading captions, maintaining consistency of annotations between similar images etc.

Step2: Parallel Annotations coupled with expert auditing and feedback: Post training, we preloaded all the images in our UI along with captions for a given topic. The images whose captions had certain important keywords were highlighted in yellow so that the crowd could prioritize annotating these images. Two annotators were asked to provide annotations in parallel, and the disagreements were tracked during the annotation process. The expert used the disagreement rate between annotators and their deviation from known ground truth, to coach the crowd so that they could continuously learn, improve and remain diligent.

Step3: Reassessment of Complex / Disagreed annotations to ensure quality: The disagreements between two annotators were sent to the 3rd annotator, and a majority opinion was taken as the final annotation. If each of the 3 annotators gave a different answer, it was sent to the expert.

5.3 Validation experiments

In this section, we aim to test whether some of our hypotheses related to advantages of 'closed crowd' hold true in the IRAT task. We believe that IRAT is a relatively simple task for the crowd compared to TRAT, and having studied value - add of our 'Skierarchy' approach in TRAT, we did not reassess it here. We tested the following hypothesis and optimized our process based on the learnings from this experiment.

Does the 'fast prototyping cycle' enabled by a closed crowd environment improve productivity without compromising on quality?

Our ability to conduct quick experiments in our 'closed crowd' setting has helped us arrive at the optimal redundancy for our process, while also providing us with the flexibility to adapt our process dynamically. For example, we were able to reduce our average redundancy from a typical industry standard of 5 annotations per image to 2.2 per image without compromising on quality. This was based on our experiment in which we introduced captions with our images and collected 2 independent reviews on 200 images from a topic. We observed that 88% times the labels agreed and they were correct. Observing this high accuracy, we decided to take a third opinion only on the remaining 12% of the articles, limiting the redundancy to maximum of 3, and a mode of 2 reviews.

5.4 Our Results on IRAT

Our relevance annotations were found to be highly accurate on IRAT as well –

	LAM	AUC
SetuServ	0.092	0.873
UT Austin	0.227	0.529

The average redundancy was 2.2 per image. Cost per label (including this redundancy) was $0.02. The average time per each label was nearly 6 seconds.

6 Related Work

A closely related area to Crowdsourcing is Human Computation that concerns with solving tasks using both humans and computers in an interactive manner. The general approach of this field is to break down the tasks into distinct tasks that humans can solve and those that computers can solve, so that the entire problem is solved using a

symbiotic interaction of humans and computers. Digitization of scanned books using ReCAPCHA (von Ahn L. M., 2008), Protein folding structure discovery using the FoldIt (Christopher B Eiben, 2012), Gamification of micro-tasks (von Ahn L. K., 2006), etc. are some examples of this technique. Note that our novel approach is related to the idea of Human Computation but is still distinct from the latter since, in our case the tasks are not broken down between computers and humans. Computers do the same tasks as humans, but humans are placed in the driver's seat with computers serving as assistants.

In the area of crowdsourcing, some work combining it with Machine Learning is beginning to emerge. Companies such as *SpeakerText.com* are using humans to correct the machine predictions of speech transcriptions. In (Vamshi Ambati, 2010), the authors proposed using Active Machine Learning approach combined with crowdsourcing to enable automatic translation for low-resource language pairs. A paper that comes closest to our work is that of (Alexander J. Quinn, 2010), in which the authors discuss a process called *CrowdFlow* that facilitates interaction between humans and machines in solving a micro-task. They present an example problem of human detection in photographs where the humans correct the bounding boxes predicted by the machine to deliver the final output. We believe ours is the first set of experiments that ties together Machine Learning as a personal assistant to humans in multiple ways including prioritization of data by difficulty, automatic highlighting of key-words, and human error correction.

7 Conclusions

Typical sources for error in crowdsourcing can be decomposed as follows: 1) Lack of competency and lack of calibration/clarification 2) Lack of due diligence 3) Subjectivity 4) Human error due to "flow bias"[11] and genuine human error. Current crowdsourcing approaches rely on micro tasking and qualification to address the first source, and ground truth and redundancy to address the next three sources of errors. While these techniques are helpful for simpler and intuitive tasks, complex tasks require pulling additional levers to control for quality. In addition to micro tasking and qualification, skill enhancement of crowd through training, expert coaching/feedback and appropriate incentives can address the first two sources of errors effectively. While redundancy is a helpful tool, it can be applied more creatively – using it where it matters the most and sometimes, using it across crowd and experts. Crowd is often good at telling where they could be going wrong; this coupled with error detection using ML and certain simple rules can help improve the quality. Also, as we proved in our experiments, for "edge" cases, redundancy with expert is much more valuable than redundancy across the crowd.

8 Bibliography

Alexander J. Quinn, B. B. (2010). CrowdFlow: Integrating Machine Learning with Mechanical Turk for Speed-Cost-Quality Flexibility. *Technical Report, University of Maryland.*

Aniket Kittur, E. H. (2008). Crowdsourcing User Studies With Mechanical Turk. *ACM CHI.*

Bernstein, M. L. (2010). Soylent: A Word Processor with a Crowd Inside. *UIST.*

Christopher B Eiben, J. B. (2012). Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature Biotechnology , 30*, 190-192.

Greg Little, L. B. (2010). Exploring Iterative and Parallel Human Computation Processes. *Human Computation.*

Vamshi Ambati, S. V. (2010). Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation (LREC)*, (pp. 2169-2174).

von Ahn, L. K. (2006). Verbosity: A Game for Collecting Common-Sense Facts. *ACM CHI.*

von Ahn, L. M. (2008). reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* , 1465-1468.

[11] This is our nomenclature for the type of error that is caused by the "momentum" of the annotator, due to which the annotator tends to be biased towards one of the ratings more than the rest.