Web Science Course

Topic Crowdsourcing

Report based on paper:

Soylent: A Word Processor with a Crowd Inside
AND
Learning From Crowds

Chenyu He
“Soylent: A Word Processor with a Crowd Inside”
Outline

- Related work
- Soylent
- Find-Fix-Verify pattern
- Evaluate the feasibility
- Conclusion
Related work

Crowdsourcing

➢ Gathering data to train algorithms

➢ Mechanical Turk used to collect labeled data for machine vision and natural language processing

➢ Having different systems to do the on-demand tasks
Artificial Intelligence for Word Processing.

• Automatic proofreading
It has a long history of research and has seen successful deployment in word processors

• Microsoft Word
Microsoft Word’s spell checker frequently suffers from false positives, particularly with proper nouns and unusual names.

• AI techniques
AI for end-user programming, allow users to demonstrate repetitive editing tasks for automatic execution.
Soylent

A prototype crowdsourced word processing interface

- Shortening
- Proofreading
- The Human Macro
Text Shortening—Soylent’s Shortn interface allows authors to condense sections of text.

- User selects the too long area of text and press start.
- Soylent launches a series of Mechanical Turk tasks in the background and notifies the user when the text is ready.
- User can continuously adjust the length of the paragraph.
Figure 1. ShortUp allows users to adjust the length of a paragraph via a slider. Red text indicates locations where cuts or rewrites have occurred. Tick marks represent possible lengths, and the blue background bounds the possible lengths.
Proofreading—Soylent provides a human-aided spelling, grammar and style checking interface called Crowdproof

- Aims to catch spelling, style and grammar errors that AI algorithms cannot find or fix.
- The process finds errors, one to five alternative rewrites.
- It is essentially a distributed proofreader operating for cents per task.
➢ highlights a section of text and presses the proofreading button
➢ the task is queued to the Soylent status pane and the user is free to keep working
➢ Soylent calls out the edited sections with a purple dashed underline when it finished

While GUIs made computers more intuitive and easier to learn, they didn’t let people be able to control computers efficiently. 

Figure 2. Crowdproof is a human-augmented proofreader. The drop-down explains the problem (blue title) and suggests fixes (gold selection).
The Human Macro: Natural Language Crowd Scripting

It is Soylent’s natural language command interface, that users can use it to request arbitrary work quickly in human language.

Figure 3. The Human Macro is an end-user programming interface for automating document manipulations. The left half is the user’s authoring interface; the right half is a preview of what the Turker will see.
Find-Fix-Verify

Challenges in Programming with Crowd Workers

➢ High Variance of Effort

Lazy Turker: does as little work as necessary to get paid.

Eager Beaver: go beyond the task requirements in order to be helpful, but create further work for the user in the process.

➢ Turkers Introduce Errors
The Find-Fix-Verify Pattern

one method of programming crowds to reliably complete open-ended tasks that directly edit the user’s data. And it can control the efforts of both the Eager Beaver and Lazy Turker and limit introduction of errors.

➢ Find-Fix-Verify: the pattern separates open-ended tasks into three stages

- **Find stage**: asks Turkers to identify patches of the user’s work that need more attention
- **Fix stage**: recruits workers to revise an identified patch
- **Verify stage**: performs quality control on revisions
Figure 4. Find-Fix-Verify identifies patches in need of editing, recruits workers to fix the patches, and votes to approve work.
Find-Fix-Verify in Soylent

Both Shortn and Crowdproof use the Find-Fix-Verify pattern

Find Stage
- Asks 10 Turkers to identify candidate areas
  - 20% of the Turkers agree on a text region

Fix Stage
- 5 Turkers see the highlighted paragraph
  - Votes whether the patch could be cut entirely

Verify Stage
- 5 Turkers see a list of all annotated rewrites
  - Votes on whether the text can be cut, if the patch can be cut, introduce the empty string as a rewrite

A set of candidate areas
A set of verified rewrites for each patch

Splitting the input region into paragraphs

Use a 15-minutes timeout at each stage to keep the algorithm responsive.
Require a minimum of 6 workers in Find, 3 in Fix, and 3 in Verify.
Evaluate the feasibility

➢ Shortn Evaluation

• Evaluate Shortn quantitatively by running on example texts.

• The goal is to see how much Shortn could shorten text, as well as its associated cost and time characteristics.

• Collecte five examples of texts that might be sent to Shortn, each between one and seven paragraphs long. Ch ose these inputs to span from preliminary drafts to finish essays and from easily understood to dense technical material (Table I).

• To simulate a real-world deployment, run the algorithms with a timeout enabled and set to 20 minutes. Requir e 6–10 workers to complete the Find tasks and 3–5 to the Fix and Verify tasks: if a Find task fails to recruit ev en six workers, it may wait indefinitely. Pay $0.08 per Find, $0.05 per Fix, and $0.04 per Verify on Mechanica l Turk.
### Table I

Our evaluation run of Shortn produced revisions between 78% – 90% of the original paragraph length on a single run. The Example Output column contains example edits from each input.

<table>
<thead>
<tr>
<th>Input</th>
<th>Original Length</th>
<th>Final Length</th>
<th>Turk Statistics</th>
<th>Time per Paragraph</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blog</td>
<td>3 paragraphs</td>
<td>83%</td>
<td>$4.57 character</td>
<td>46 – 57 min</td>
<td>Print publishers are in a tizzy over Apple’s new iPad because they hope to finally be able to charge for their digital editions. But in order to get people to pay for their magazine and newspaper apps, they are going to have to offer something different that readers cannot get at the newsstand or on the open Web.</td>
</tr>
<tr>
<td>Classic UIST [28]</td>
<td>7 paragraphs</td>
<td>87%</td>
<td>$7.45 character</td>
<td>49 – 84 min</td>
<td>The metaDESK effort is part of the larger Tangible Bits project. The Tangible Bits vision paper, which introduced the metaDESK along with two companion platforms, the transBOARD and ambientROOM.</td>
</tr>
<tr>
<td>Draft UIST [29]</td>
<td>5 paragraphs</td>
<td>90%</td>
<td>$7.47 character</td>
<td>52 – 72 min</td>
<td>In this paper we argue that it is possible and desirable to combine the easy input affordances of text with the powerful retrieval and visualization capabilities of graphical applications. We present WenSo, a tool that uses lightweight text input to capture richly structured information for later retrieval and navigation in a graphical environment.</td>
</tr>
<tr>
<td>Rambling E-mail</td>
<td>6 paragraphs</td>
<td>78%</td>
<td>$9.72 character</td>
<td>44 – 52 min</td>
<td>A previous board member, Steve Burleigh, created our website last year and gave me a lot of ideas. For this year, I found a website called eTeamIZ that hosts websites for sports groups. Check out our new page: [...]</td>
</tr>
<tr>
<td>Technical Comp. Sci. [3]</td>
<td>3 paragraphs</td>
<td>82%</td>
<td>$4.84 character</td>
<td>132 – 489 min</td>
<td>Figure 3 shows the pseudocode that implements this design for Lookup. FAWN-DS extracts two fields from the 160-bit key: the 15 low order bits of the key (the index bits) and the next 15 low order bits (the key fragment).</td>
</tr>
</tbody>
</table>
Crowdproof Evaluation

• Obtain a set of five input texts in need of proofreading (Table II).

• Manually label all spelling, grammatical and style errors in each of the five inputs, identifying a total of 49 errors.

• Run Crowdproof on the inputs using a 20-minute stage timeout, with prices $0.06 for Find, $0.08 for Fix, and $0.04 for Verify.

• Measure the errors that Crowdproof catches, that Crowdproof fixes, and that Word catch.

• Rule that Crowdproof has caught an error if one of the identified patches contain the error.
### Table II. A report on Crowdproof's runtime characteristics and example output.

<table>
<thead>
<tr>
<th>Input</th>
<th>Content</th>
<th>Errors all/case/shifted</th>
<th>Turkers</th>
<th>Time</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Web Science</strong></td>
<td><strong>Investigating the Future of Information and Communication</strong></td>
<td></td>
<td></td>
<td></td>
<td>Marketing seems bad for brands big and small. You know what I am saying. It is no wonder that advertising seems bad for companies in America, Chicago, and Germany. Updating of brand image seems bad for processes in one company and many companies.</td>
</tr>
<tr>
<td>Notes</td>
<td>2 paragraphs 8 sentences 107 words</td>
<td>14 / 8 / 8</td>
<td>$4.72</td>
<td>42-53</td>
<td>Blah blah blah—This is an argument about whether there should be a standard &quot;nosql NoSQL storage&quot; API to protect developers storing their stuff in proprietary services in the cloud. Probably unrealistic. To protect yourself, use an open software offering, and self-host or go with hosting solution that uses open offering.</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1 paragraph 5 sentences 63 words</td>
<td>8 / 7 / 6</td>
<td>$2.18</td>
<td>54</td>
<td>Dandu Monara (Flying Peacock, Wooden Peacock), The Flying Machine able to fly. The King Ravana (Sri Lanka) built it. According to Hindu believers in Ramayana, King Ravana used &quot;Dandu Monara&quot; for abducted queen Seetha from Rama. According to believers, &quot;Dandu Monara&quot; landed at Werangatota.</td>
</tr>
<tr>
<td>UIST Draft</td>
<td>1 paragraph 6 sentences 135 words</td>
<td>6 / 4 / 3</td>
<td>$3.30</td>
<td>96</td>
<td>Many of these problems vanish if we turn to a much older recording technology—text. When we enter text, each (pen or key) stroke is being used to record the actual information we care about—none is wasted on application navigation or configuration.</td>
</tr>
</tbody>
</table>
Human Macro Evaluation

Interest in understanding whether end users could instruct Mechanical Turk workers to perform open-ended tasks. Can users communicate their intention clearly? Can Turkers execute the amateur-authored tasks correctly?

• Generate 5 feasible Human Macro scenarios (Table III).

• Recruit two sets of users: 5 undergraduate and graduate students (4 male) and 5 administrative associates (all female).

• Show each user one of the five prompts, consisting of an example input and output pair.

• Introduce participants to the Human Macro, ask them write task description, then send the request to MTurk

• Results in two quality metrics: intention and accuracy
<table>
<thead>
<tr>
<th>Task</th>
<th>Quality</th>
<th>Example Request</th>
<th>Example Input</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tense</td>
<td>CS: 100% intention, (20% accuracy)</td>
<td>Admin: “Please change text in document from past tense to present tense.”</td>
<td>I gave one final glance around before descending from the barrow. As I did</td>
<td>I give one final glance around before descending from the barrow. As I do</td>
</tr>
<tr>
<td></td>
<td>1 paragraph</td>
<td></td>
<td>so, my eye caught something […]</td>
<td>so, my eye catches something […]</td>
</tr>
<tr>
<td>Figure</td>
<td>CS: 75% (75%)</td>
<td>CS: “Pick out keywords from the paragraph like Yosemite, rock, half</td>
<td>When I first visited Yosemite State Park in California, I was a boy. I was</td>
<td><a href="http://commons.wikimedia.org">http://commons.wikimedia.org</a></td>
</tr>
<tr>
<td></td>
<td>1 paragraph</td>
<td>dome, park. Go to a site which has CC licensed images […]”</td>
<td>amazed by how big everything was […]</td>
<td>/wiki/File:03_yosemite_half_dome.jpg</td>
</tr>
<tr>
<td>Opinions</td>
<td>CS: 100% (100%)</td>
<td>CS: “Please tell me how to make this paragraph communicate better. Say what’s</td>
<td>Take a look at your computer. Think about how you launch programs, edit</td>
<td>This paragraph needs an objective I</td>
</tr>
<tr>
<td></td>
<td>1 paragraph</td>
<td>wrong, and what I can improve. Thanks!”</td>
<td>documents, and browse the web. Don’t you feel a bit lonely? […]</td>
<td>feel like. […] After reading I feel like there should be about five more</td>
</tr>
<tr>
<td>Citation</td>
<td>CS: 75% (75%)</td>
<td>Admin: “Hi, please find the bibtex references for the 3 papers in brackets.</td>
<td>Duncan and Watts [Duncan and Watts HCOMP 09 anchoring] found that Turkers</td>
<td>sentences […]</td>
</tr>
<tr>
<td>Gathering</td>
<td>Admin: 100% (100%)</td>
<td>You can located these by Google Scholar searches and clicking on bibtex.”</td>
<td>will do more work when you pay more, but that the quality is no higher.”]</td>
<td>@conference{ title={Financial incentives and […]}, authors={Mason, W. and</td>
</tr>
<tr>
<td></td>
<td>1 paragraph</td>
<td></td>
<td></td>
<td>Watts, D.J.}, booktitle={HCOMP ’09}</td>
</tr>
<tr>
<td>List</td>
<td>CS: 82% (82%)</td>
<td>Admin: “Please complete the addresses below to include all information needed</td>
<td>Max Marcus. 3416 colfax ave east, 80206</td>
<td>Max Marcus</td>
</tr>
<tr>
<td>Processing</td>
<td>Admin: 98% (96%)</td>
<td>as in example below. […]”</td>
<td></td>
<td>3416 E Colfax Ave Denver, CO 80206</td>
</tr>
<tr>
<td></td>
<td>10 inputs</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table III. The five tasks in the left column led to a variety of request strategies. Tense, error-filled user requests still often led to success.
Conclusion

- Present Soylent, a word processing interface that enables writers to call on Mechanical Turk workers.

- To shorten, proofread, and otherwise edit parts of their documents on demand.

- To improve worker quality, we introduce the Find-Fix-Verify crowd programming pattern, which splits tasks into a series of generation and re-view stages.

- Evaluation studies demonstrate the feasibility of crowdsourced editing and investigate questions of reliability, cost, wait time, and work time for edits.
“Learning From Crowds”
Outline

- Basic Concept
- Experimental Validation
- Conclusions
1. Basic Concept

1.1 Supervised Learning From Multiple Annotators/Experts

- **The Problem With Majority Voting**

- use the labels on which the majority of them agree (or average for regression problem) as an estimate of the actual gold standard, that is:

\[
\hat{y}_i = \begin{cases} 
1 & \text{if } (1/R) \sum_{j=1}^{R} y_{ij} > 0.5 \\
0 & \text{if } (1/R) \sum_{j=1}^{R} y_{ij} < 0.5
\end{cases}
\]

- considering every pair (instance, label) provided by each expert as a separate example, that is:

\[
\Pr[y_i = 1|y_{i1}^1, \ldots, y_{iR}^R] = \frac{1}{R} \sum_{j=1}^{R} y_{ij}.
\]
1.2 Binary Classification

• The EM(Expectation-Maximization) Algorithm

The EM algorithm is an efficient iterative procedure to compute the maximum-likelihood solution in presence of missing/hidden data.

Use the unknown hidden true label $y_i$ as the missing data. If we know the missing data $y = [y_1, \ldots, y_N]$ then the complete likelihood can be written as:

$$\ln \Pr[D, y|\theta] = \sum_{i=1}^{N} y_i \ln p_i a_i + (1 - y_i) \ln (1 - p_i) b_i.$$ 

- Each iteration of the EM algorithm consists of two steps: Expectation(E)-step and Maximization(M)-step. The M-step involves maximization of a lower bound on the log-likelihood that is refined in each iteration by the E-step.

- These two steps (the E- and the M-step) can be iterated till convergence. The log-likelihood increases monotonically after every iteration, which in practice implies convergence to a local maximum. The EM algorithm is only guaranteed to converge to a local maximum.
2. Experimental Validation

2.1 Classification Experiments

- Use two CADs and one text data set in experiments. The CAD datasets include a digital mammography dataset and a breast MRI dataset, both of which are biopsy proven that the gold standard is available.

2.1.1 Digital Mammography with simulated Radiologists

- Mammograms are used as a screening tool to detect early breast cancer. CAD systems search for abnormal areas (lesions) in a digitized mammographic image.

- In this experiment we use a proprietary biopsy-proven data set (Krishnapuram et al., 2008) containing 497 positive and 1618 negative examples.
The following observations can be made:

- **Classifier performance:** Figure 1(a) plots the ROC curve of the learnt classifier on the training set.
Radiologist performance: The actual sensitivity and specificity of each radiologist is marked as a black × in Figure 1(b).
Actual ground truth: Since the estimates of the actual ground truth are probabilistic scores, we can also plot the ROC curves of the estimated ground truth. From Figure 1(b) we can see that the ROC curve for the proposed method dominates the majority voting ROC curve; The improvement obtained is quite large in Figure 2 which corresponds a situation where we have only one expert and 7 novices.
**Joint Estimation:** To learn a classifier, Smyth et al. (1995) proposed to first estimate the golden ground truth and then use the probabilistic ground truth to learn a classifier. *Figure 3* shows that the classifier and the ground truth learnt obtained by the proposed algorithm is superior than that obtained by other procedures which first estimates the ground truth and then learns the classifier.

![ROC Curve for the classifier](image1.png)

![ROC Curve for the estimated true labels](image2.png)

*Figure 3:* ROC curves comparing the proposed algorithm (solid red line) with the *Decoupled Estimation* procedure (dotted blue line), which refers to the algorithm where the ground truth is first estimated using just the labels from the five radiologists and then a logistic regression classifier is trained using the soft probabilistic labels. In contrast, the proposed EM algorithm estimates the ground truth and learns the classifier simultaneously during the EM algorithm.
2.1.2 Breast MRI

- In this example, each radiologist reviews the breast MRI data and assesses the malignancy of each lesion on a BIRADS scale of 1 to 5. The BIRADS scale is defined as follows: 1 Negative, 2 Benign, 3 Probably Benign, 4 Suspicious abnormality, and 5 Highly suggestive of malignancy. The dataset comprises of 75 lesions with annotations from four radiologists, and the true labels from biopsy.

The confusion matrix in Table 1 shows that the EM algorithm is significantly superior than the majority voting in estimating the true BIRADS.

Table 1: The confusion matrix for the estimate obtained using majority voting and the proposed EM algorithm. The x indicates that there was no such category in the true labels (the gold standard). The gold-standard is obtained by the biopsy which can confirm whether it is benign (BIRADS=2) or malignant (BIRADS=5).
Figure 4 summarizes the results. The leave-one-out cross validated ROC for the classifier. The cross-validated AUC of the proposed method is approximately 6% better than the majority voting baseline.
2.1.3 Recognizing Textual Entailment

- Report results on Recognizing Textual Entailment data collected by Snow et al. (2008) using the Amazon’s Mechanical Turk. In this task, the annotator is presented with two sentences and given a choice of whether the second sentence can be inferred from the first. The data has 800 tasks and 164 distinct readers, with 10 annotations per task along with the golden ground truth.

- Figure 5 plots the accuracy of the estimated ground truth as a function of the number of annotators. The proposed EM algorithm achieves a higher accuracy than majority voting. In other words to achieve a desired accuracy the proposed algorithm needs fewer annotators than the majority voting scheme.
2.2 Regression Experiments
- Explain the algorithm on a toy dataset and then present a case study for automated polyp measurements.

2.2.1 Illustration

» Figure 6 shows the proposed algorithm for regression on a one-dimensional toy data set with three annotators. The actual regression model (shown as a blue dotted line) is given by $y = 5x - 2$. Simulate 20 samples from three annotators with precisions 0.01, 0.1, and 1.0. The data are shown by the annotator’s number.
2.2.2 Automated Polyp Measurements

Colorectal polyps are small colonic findings that may develop into cancer at a later stage. The diameter of the polyp is one of the key factors which decides the malignancy of a suspicious polyp. Hence accurate size estimation is crucial to decide the action to be taken on a polyp.

Use a proprietary data set containing 393 examples (which point to 285 distinct polyps—the segmentation algorithms generally return multiple marks on the same polyp.) along with the measured diameter (ranging from 2mm to 15mm) as the training set. Each example is described by a set of 60 morphological features which are correlated to the diameter of the polyp.

Figure 7 shows the scatter plot of the actual polyp diameter vs the diameter predicted by the three different models. Compare the performance based on the root mean squared error (RMSE) and also the Pearson’s correlation coefficient. The Regressor learnt using the proposed iterative algorithm (Figure 7(b)) is almost as good as the one learnt using the golden ground truth (Figure 7(a)).
Figure 7: Scatter plot of the actual polyp diameter vs the diameter predicted by the models learnt using (a) the actual gold standard, (b) the proposed algorithm with annotations from five radiologists, and (c) the average of the radiologist’s annotations. (See § 6.2.2 for a description of the experimental setup.)
3. Conclusions

- Proposed a probabilistic framework for supervised learning with multiple annotators providing labels but no absolute gold standard.

- The proposed algorithm iteratively establishes a particular gold standard, measures the performance of the annotators given that gold standard, and then refines the gold standard based on the performance measures.

- Discussed binary/categorical/ordinal classification and regression problems.
Thank you