Crowdsourcing Similarity Judgments for Agreement Analysis in End-User Elicitation Studies

Abdullah X. Ali
The Information School
DUB Group
University of Washington
Seattle, WA 98195 USA
xyleques@uw.edu

Meredith Ringel Morris
Microsoft Research
Redmond, WA, 98052 USA
merrie@microsoft.com

Jacob O. Wobbrock
The Information School
DUB Group
University of Washington
Seattle, WA 98195 USA
wobbrock@uw.edu

ABSTRACT

End-user elicitation studies are a popular design method, but their data require substantial time and effort to analyze. In this paper, we present *Crowdsensus*, a crowd-powered tool that enables researchers to efficiently analyze the results of elicitation studies using subjective human judgment and automatic clustering algorithms. In addition to our own analysis, we asked six expert researchers with experience running and analyzing elicitation studies to analyze an end-user elicitation dataset of 10 functions for operating a web-browser, each with 43 voice commands elicited from end-users for a total of 430 voice commands. We used Crowdsensus to gather similarity judgments of these same 430 commands from 410 online crowd workers. The crowd outperformed the experts by arriving at the same results for seven of eight functions and resolving a function where the experts failed to agree. Also, using Crowdsensus was about four times faster than using experts.

Author Keywords

End-user elicitation study; agreement rate; online crowds; crowdsourcing; human computation; Mechanical Turk.

INTRODUCTION

End-user elicitation studies are a popular design method. They have been used to generate commands for command line interfaces [10], to design gesture interactions [20,26,32], and in virtual reality [15]. Wobbrock *et al.* [37,38] formalized a method to conduct gesture elicitation studies in the lab. The method works by presenting the effect of an interaction to participants and eliciting the action or symbol meant to invoke that effect. Once researchers have collected action or symbol proposals from end users, they resolve these into representative sets by grouping them based on similarity.

By having end users propose interactions, elicitation studies aim to create interaction designs that are more intuitive, *i.e.*, interactions that may be more discoverable, learnable, guessable, memorable, or comfortable. Larger and more diverse sets of participants can improve the intuitiveness of the final set of interaction designs [22].

Despite the popularity of end-user elicitation studies, having been conducted in more than 60 published accounts (*e.g.*, [23,25,27,32]), such studies are laborious to run and analyze, especially grouping elicited proposals based on their similarity. Although including a large and diverse group of end users in elicitation studies is desirable, this practice vastly increases the number of comparisons among proposals when analyzing the results of elicitation studies, increasing the workload. In this paper, we propose using a combination of online crowds and automatic clustering algorithms to determine the similarity of elicited proposals from elicitation studies, thereby eliminating the burden of manually comparing elicited proposals.

To support efficient elicitation study analysis, we created *Crowdsensus*. Crowdsensus generates web interfaces that present crowd workers with interaction design proposals collected in an elicitation study and asks them to vote on their similarity. After collecting the votes, Crowdsensus employs automatic clustering algorithms to find agreement between proposals using the votes, thereby resolving the underlying set of proposals. We used the Crowdsensus system to explore the following three research questions:

Q1. How can the crowd facilitate similarity judgments for agreement analysis in end-user elicitation studies?

Q2. By using the crowd, what are the benefits, if any, in terms of cost and time compared to the status quo use of experts' judgments?

Q3. How does the quality of the results produced by the crowd compare to those produced by expert researchers?

In pursuing answers to these questions, we address various challenges, including determining the design of the interface the crowd workers should interact with, the number of symbols that should be presented, the best phrasing of the

UIST '18, October 14–17, 2018, Berlin, Germany
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5948-1/18/10…$15.00
https://doi.org/10.1145/3242587.3242621

instructions, how to detect spam answers, and which clustering algorithms to employ.

We deployed a study on Amazon's Mechanical Turk platform to validate our approach. In the study, 410 crowd workers were asked to find the similarity among 43 proposed text representations of voice commands for 10 functions of a voice-activated web browser, for a total of 430 voice command proposals. We found that a crowd of non-expert workers, in combination with automatic clustering algorithms, was able to select the same commands for seven out of eight prompts (out of 10 total) for which a group of experts' judgments converged. As for the two prompts (out of 10) for which the experts' opinions diverged, the crowd successfully converged upon a command for one of them. Also, the crowd was about four times faster than the expert researchers.

This work contributes a method to crowdsource the analysis of end-user elicitation studies, cutting effort and time. We also contribute the Crowdsensus system itself,[1] a tool for researchers to utilize online crowds to find agreement in complex datasets (or analyze data from elicitation studies themselves). Crowdsensus can be used equally well to find agreement among gestures, icon sketches, text commands, or voice commands—any situation where symbolic inputs can be used to invoke commands.

ELICITATION STUDIES: A BRIEF INTRODUCTION

For context, we describe elicitation studies, how to conduct them, their benefits and drawbacks, as well as some terminology that appears throughout this paper.[2] For illustrative purposes, we discuss the hypothetical example of a researcher attempting to design gesture controls for a smartphone file-explorer that requires interactions to perform the following functions: open a file, close a file, and delete a file.

Why Run an Elicitation Study?

The researcher's goal is to design touch screen gestures for each of her file-explorer app's functions that feel natural and intuitive to users. The researcher hopes that running an elicitation study will allow her to elicit intuitive gesture designs from her target population. Also, the gesture designs proposed by users might be ones that the researcher herself had not imagined. Such a study also allows her to identify synonyms and variations for gesture designs.

How to Run an Elicitation Study?

There are two parts to every elicitation study: data gathering and data analysis. In the first part, the researcher invites end users to her lab. She shows the participants the results of an action, known as a *referent* [37]. In this example, the referents shown are: opening a file, closing a file, and deleting a file. The researcher asks each participant to take an action that would cause each referent to occur. The proposed interaction designs in these studies are typically referred to as *symbols* [37]. Symbols can be any form of interaction between a user and the technology they are using; typically, the researcher will specify to participants what forms of symbols the target technology can recognize. For example, the researcher might instruct participants to only propose symbols that are text strings for command line interfaces, audio clips for voice user interfaces, or sketches of icons for graphical user interfaces.

Once the researcher has collected enough symbols for her referents, she moves on to the second part of the study: analyzing the symbols. For each referent, the researcher has a set of symbols—touch gestures in this example—collected from the participants. The researcher compares all of the symbols for a given referent to each other and groups them based on their similarity. After grouping the symbols, the researcher calculates an agreement score, which quantifies the consensus among participants. The original formula [37] for calculating agreement is:

$$\mathrm{A} = \frac{1}{|\mathrm{R}|}\sum_{\mathrm{r}\in\mathrm{R}}\sum_{\mathrm{P_i}\subseteq\mathrm{P_r}}\left(\frac{|\mathrm{P_i}|}{|\mathrm{P_r}|}\right)^2 \qquad (1)$$

In Eq. 1, r is a referent in the set of all referents R. P_r is the set of all symbols proposed for referent r. P_i is a subset of similar symbols in P_r. Subsequent variations to this formula have been published but all are similar [9,33,34].

The researcher uses agreement calculations to determine if the largest set of similar symbols have sufficient membership to be a singularly good choice, or if multiple symbols should be used synonymously for the same referent. Another use of the agreement calculation is to resolve conflict in cases where the same symbol is proposed for multiple referents. In that case, the symbol usually is assigned to the referent for which that symbol has the highest agreement score.

Challenges of Elicitation Studies

As mentioned above, the more symbols that are elicited from more participants, the more representative of users' behavior the ultimate gesture interactions for each function should be [22]. Nevertheless, as the number of symbols increases, the number of similarity comparisons the researcher has to do increases significantly. Unless the elicited symbols can be automatically compared, like integer comparisons (*e.g.*, [37]), subjective human judgment (and labor) is needed to group symbols by similarity. For example, if three participants pinched their thumb and index finger together for "close a file" and two participants pinched their thumb and middle finger together instead, the researcher would have to decide whether these actions were two distinct gestures (because of the variation in finger used) or should be considered one similar proposal (touch the thumb to a finger). This decision might be influenced by other factors, such as the form of the other proposals in the gesture set, or

[1] http://depts.washington.edu/madlab/proj/crowdsensus/

[2] Readers wishing to know more about end-user elicitation studies are encouraged to read some of the pivotal papers [21,22,37,38].

the limitations of the relevant technology platform (*e.g.*, the ability to detect different fingers). Further, this process is time-consuming, as the researcher may have to watch videos (in the case of gestures) many times to assess the nature of a particular symbol and its similarity to other symbols.

In addition to the subjective nature of the work required to analyze these studies, comparing symbols is taxing both in time and effort. In our example, if the researcher recruited 30 participants, and each one of the participants proposed only two symbols (here, gestures) for each of the researcher's three referents (here, app file functions), then for each referent, the researcher would have to compare 60 symbols to each other, for a total of 3600 comparisons per referent. That's 10,800 comparisons for all three referents. The amount of time and effort required by the researcher is the main problem our work is addressing. We propose the use of online crowd workers in combination with automatic clustering algorithms to achieve the desired result of symbol grouping for referents.

RELATED WORK

Relevant prior work includes studies eliciting user input, methodological improvements to end-user elicitation studies, and the use of online crowds in HCI**.**

End-User Elicitation Studies

The practice of employing users to propose interaction designs is not new, with the earliest example dating back to 1984 when Good *et al.* [10] had users propose command terms to design an intuitive command-line interface. Wobbrock *et al.* [35,36] formalized their gesture elicitation method using stylus gestures and, later, tabletop hand gestures, adding conflict resolution and agreement calculations. Many have replicated Wobbrock *et al.*'s methodology in HCI research; for example, Morris *et al.* [23] used it to elicit 357 speech and gesture interactions for TV-based web browsing, Obaid *et al.* [25] used the method to elicit 385 body gestures for controlling humanoid robots, and Piumsomboon *et al.* [27] used it to explore 800 user-defined interactions for augmented reality. Our work aims to reduce the resources required to analyze the results of such studies while allowing researchers to conduct elicitation studies on a large scale.

Improvements to Elicitation Methods

Given the popularity of the end-user elicitation method, there has been a considerable body of work that contributed extensions to it, both in the elicitation step and in the agreement analysis step. Morris *et al.* [21] introduced principles to reduce legacy bias in end users when proposing symbols such as gestures. Vatavu and Wobbrock [33,34] added new agreement calculations, disagreement scores, and qualitative judgments; Morris [23] also proposed consensus metrics to supplement agreement scores. Other work [24,30] made contributions to gesture-specific analysis. Our work contributes a tool to reduce the time and effort required to analyze elicitation studies regardless of the type of symbols collected, whether the researchers are doing the analysis themselves or employing a crowd of online workers to provide the subjective similarity judgments for them.

Online Crowds in HCI Research

The use of an online crowd of workers, paid or unpaid, to accomplish tasks that are computationally difficult or impossible is a common practice [4,6,19]. Bragg and Weld [5] and Chilton *et al.* [7] created algorithms to enable crowd workers to generate taxonomy labels for items such as images. Tamuz *et al.* [33] and others [12,13,28] used crowds to cluster data using the triplet model, an approach different than ours. Our work aims to use crowd workers to provide subjective similarity judgments for agreement analysis in end-user elicitation studies, where the symbols the crowd is judging can be complex, like freehand gestures, icon sketches, or audio clips, not just strings of text.

CROWDSENSUS: A SIMILARITY JUDGEMENT PLATFORM

We developed a tool called *Crowdsensus* to capture and analyze symbol-similarity votes from online crowds. Crowdsensus generates custom web interfaces that facilitate the collection of similarity votes from online crowd workers. The interfaces are platform-agnostic and run on any device with a web browser.

Importing Symbols

After conducting an elicitation study, a Crowdsensus user uploads a set of symbols collected for a referent. Currently, this set takes the form of a comma separated value file (CSV).[3] The symbols are stored in the Crowdsensus database to be used in generating similarity-judgment webpages for the uploaded data.

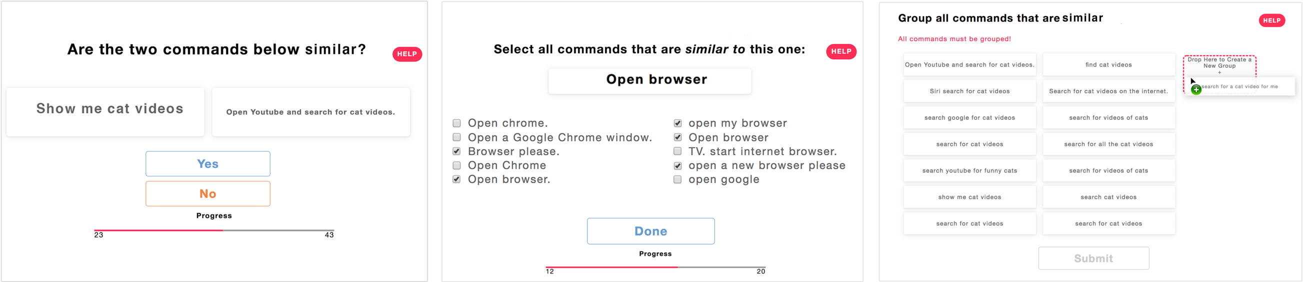
Interface Designs for Similarity Judgments

Crowdsensus first shows an instruction page that includes a training video on how to complete the task of similarity voting. Once crowd workers are familiar with the task, they click "Start." The main screen shows a prompt asking them to select similar symbols. There is a "Help" button that brings up the instructions screen with the training video. At the bottom, a "Done" button takes the workers to the next task, and a progress bar indicates how far through the process they are. We designed three different approaches to present the symbols to the workers for comparison, described below.

The Direct Comparison Interface "(1:1)"

The first design presented direct comparisons between two symbols, *i.e.*, a 1:1 comparison. With this approach, crowd workers see two proposed symbols for a given referent displayed side-by-side, with two buttons under them labeled "Yes" and "No."

[3] Currently, Crowdsensus uses text-based symbols and referents. Future work will extend the tool to handle other forms of interaction, such as audio clips, still images, and videos, to convey different interaction modalities.



Are the two commands below similar?

Show me cat videos

Open Youtube and search for cat videos.

A. The Direct Comparison Interface (1:1)

Select all commands that are *similar to* this one:

Open browser

- □ Open chrome.
- □ Open a Google Chrome window.
- ☑ Browser please.
- □ Open Chrome
- ☑ Open browser.
- ☑ open my browser
- ☑ Open browser
- □ TV. start internet browser.
- ☑ open a new browser please
- □ open google

B. The List Comparison Interface (1:N)

Group all commands that are similar

All commands must be grouped!

Open Youtube and search for cat videos.

find cat videos

Siri search for cat videos

Search for cat videos on the internet.

search google for cat videos

search for videos of cats

search for cat videos

search for all the cat videos

search youtube for funny cats

search for videos of cats

show me cat videos

search cat videos

search for cat videos

search for cat videos

Drop Here to Create a New Group +

search for a cat video for me

C. The Grouping Interface (N:N)

Figure 1. The three comparison interfaces in *Crowdsensus*. **(A)** The Direct Comparison interface. A prompt asks crowd workers to vote on the similarity of two commands. **(B)** The List Comparison interface. The worker selects from a list of symbols those he thinks are similar to the one highlighted in the middle of the screen. **(C)** A list of draggable symbols, with one dragged over the "create new group" drop zone.

If the workers think the two symbols are similar, they click "Yes"; otherwise, they click "No." Figure 1A shows a screenshot of the Direct Comparison interface.

The List Comparison Interface "(1:N)"

The second design was the comparison of a symbol to a list (Figure 1B), where one symbol for a given referent is being compared to a subset of the other symbols proposed for that same referent, *i.e.*, a 1:N comparison. This design shows a symbol in a box in the middle of the screen and a checklist of other proposed symbols from which to select.

The Grouping Interface "(N:N)"

The final similarity judgment interface, the Grouping interface, presents all of the proposed symbols for a given referent inside draggable elements. Crowd workers compare all symbols to each other in a many-to-many approach, *i.e.*, an N:N comparison. On the right-hand side of the screen, there is a "create a new group" drop zone. The workers have to drag-and-drop elements onto the drop zone to create a new group (Figure 1C). The workers repeat the operation, dragging-and-dropping symbols either onto established groups or to create new groups. On the right side of every group there is a count showing the number of symbols in that group. Clicking on a group will open a pop-up that displays all of the symbols belonging to that group.

Preference for the List Comparison Interface "(1:N)"

After thoroughly exploring the benefits and downsides of these three grouping interfaces, we eliminated the Direct Comparison interface (1:1) due to the large volume of tasks required to get adequate votes for analysis. For each referent there had to be N^2 1:1 comparisons, where N is the number of symbols for that referent. For instance, in the study we analyze in the next section, we gathered 43 symbols for 10 referents, which would need 18,490 comparisons to get a single similarity vote for every pair of symbols. Also, it was hard to create vote validation mechanisms for this interface to eliminate spam voting. On the flip side, we eliminated the Grouping interface (N:N) because pilot testing showed it was too complex and frustrating for the workers when working with a large dataset. Future work with smaller datasets might reconsider this design. We therefore decided to make the List Comparison interface (1:N) the default interface for Crowdsensus. By providing a list of options, the 1:N interface provides more context for the user than the 1:1 interface, but is less overwhelming than the N:N interface.

Prompts for Similarity Judgments

At the top of the List Comparison interface there is a crucial prompt instructing the crowd workers to vote on the similarity of the presented symbol to those appearing in the list. We tested four different phrasings of this prompt. The prompt was phrased with the following variations: *"Select all commands that are ('essentially the same as,' 'substantially similar to,' 'similar to,' 'kinda similar to') this one."* From pilot testing, we found that the phrase "essentially the same" and "substantially similar" yielded very strict, inflexible similarity voting from the crowd. Conversely, the phrase "kinda similar" gave very loose similarity votes, resulting in an agreement score of 1.00, meaning that the crowd voted all symbols to be "kinda the same." We found that the best prompt to use was simply, *"Select all commands that are similar to this one."* Such neutral phrasing struck a nice balance between promoting strict and permissive comparisons.

Number of Symbols Presented

In the List Comparison interface (1:N), the number of symbols listed could conceivably vary from as few as two symbols up to N, the entire set of symbols for a given referent. We tested out three different configurations of the number of symbols shown: 10 symbols, 20 symbols, and all 43 symbols that we used in the study presented in this paper, below. In our pilot testing, we found that the number of symbols presented actually did not affect the crowd's performance or our results, making it possible to present large sets of data in smaller subsets that fit on a single screen.

Vote Validation

To eliminate spam votes, we devised two procedures for vote validation: time thresholds and the identical symbol test.

Time Threshold

From pilot testing, we saw that the average time spent voting on a single List Comparison task with 43 symbols in the list was a little over 30 seconds. We decided to put a threshold of 15 seconds per task to accept valid data. Data coming from voters who did not spend at least 15 seconds looking at the symbols were disqualified and not recorded.

Identical Symbol Test

As stated, in the List Comparison interface, there was a list of symbols being compared to the primary symbol. That list included the primary symbol itself. Any answers submitted where the primary symbol itself was *not* selected were discarded, since a failure to self-match indicates the worker was not paying adequate attention to the task.

EVALUATING CROWDSENSUS: DATA COLLECTION

To answer our research questions around using online crowds to generate similarity judgments for agreement analysis in end-user elicitation studies, we created a set of referents and symbols. We then analyzed this data with two methods: manual analysis by elicitation study experts, and crowd-based analysis using Crowdsensus.

Creating a Set of Referents and Symbols

Typically, elicitation studies are carried out in a laboratory. For this study, we conducted an online elicitation study with workers on Amazon Mechanical Turk to create a set of symbols for evaluating Crowdsensus. We based the referents on Morris's "Web on the Wall" study [23]. We asked 43 crowd workers to type in text strings representing voice commands that they felt would be the most intuitive way to perform each of our 10 referents to interact with a voice-controlled web browser at a distance (see Table 4). We chose to elicit the commands in the form of text strings rather than audio clips in order to simplify the format of data capture and storage. The set of referents and symbols generated by this study is available for other researchers to use and is included as Supplementary Materials accompanying this paper.

Grouping of the Symbols by Elicitation Experts

We requested groupings of these symbols by a set of elicitation experts (the *status quo* method by which such data are analyzed) to compare the experts' output to that of the crowd via Crowdsensus. In addition, the first author of this paper analyzed the data as an expert would, using the protocol established by prior work [38]. For every one of the 10 referents, the experts grouped the elicited symbols based on their similarity, and calculated the agreement score using Eq. 1, above. For each referent, the experts elected the group with the largest number of similar symbols to trigger the given referent. Aliasing a single referent to multiple symbol synonyms was not considered for this study.

Our elicitation experts were considered experts because they had previously published research using elicitation studies. All experts had Ph.D. degrees and were external to our own university. Experts did all groupings before any output from Crowdsensus was available. Table 1 gives the experts' demographic information.

We sent each of the experts an Excel spreadsheet that included a separate tab for each of the 10 referents. Within each tab, 43 separate rows contained the text of the proposed symbols for that referent. We asked them to group the symbols for each referent by entering a group number in a column adjacent to each symbol, and to record how long it took them to complete the task using provided timing software. As compensation for their time, each expert received a $50 Amazon gift card. We chose this compensation level based on the amount of time it took the first author to analyze the same data (approximately 30 minutes), estimating that faculty are compensated at approximately $100 / hour based on typical faculty salaries in the U.S.

Grouping of the Symbols by the Crowd

We recruited 410 workers from Amazon Mechanical Turk. The majority of them were between the age of 26 and 40. Sixty-three percent of the workers had bachelor's degrees, and 85% of them considered English to be their primary language. Table 1 provides the workers' demographic information. Workers on Mechanical Turk who accepted the human intelligence task (HIT) went to a webpage on our Crowdsensus server by following the link in the HIT itself. The page provided instructions, human subjects study approval details, and researchers' contact information. Before starting the task, the workers had to fill out a brief, pre-task survey that collected demographic information, as well as details about their familiarity with voice-operated technologies.

Demographic		Experts	Crowd
Gender	Male	6 (86%)	224 (55%)
	Female	1 (14%)	186 (45%)
	Other	0	0
Age	18-25	0	155 (38%)
	26-40	6 (86%)	214 (52%)
	41-55	1 (14%)	28 (7%)
	56 and over	0	13 (3%)
Education level	Less than high school	0	3 (1%)
	Graduated high school	0	65 (16%)
	Technical school	0	14 (3%)
	Associate degree	0	38 (9%)
	Bachelor's degree	0	257 (63%)
	Advanced degree	7 (100%)	33 (8%)
Country	USA	6 (86%)	136 (33%)
	India	0	250 (61%)
	Other	1 (14%)	24 (6%)
English is primary language	Yes	4 (57%)	350 (85%)
	No	3 (53%)	60 (15%)
Frequency of voice command use	Never	1 (15%)	184 (45%)
	Daily	5 (70%)	88 (21%)
	Weekly	0	105 (26%)
	Monthly	1 (15%)	32 (8%)

Table 1. Demographic information for the academic experts, plus the first author of this paper, and the crowd workers who analyzed our web-browser-voice-command elicitation data set.

After completing the background survey, workers moved on to the page generated by the Crowdsensus tool. Pilot testing showed that workers would be able to finish 20 runs of the List Comparison interface (*i.e.*, 1:N for $N = 43$ symbols) within about 15 minutes. Therefore, we compensated the workers $2.75 per HIT. We based our pricing on the recommendation of Silberman *et al.* [31], equaling a rate of

$11/hour (our state's minimum wage). After finishing the 20 voting tasks, the participants moved on to a page that thanked them for participating and provided them with a unique code to enter back into the HIT page on Mechanical Turk. We used those unique codes to track which participants finished the HIT.

EVALUATING CROWDSENSUS: CLUSTERING

A crucial step after the crowd has provided votes indicating perceived similarity among symbols is to group these symbols. For a given referent, all pairs of symbols proposed for it have a level of similarity expressed by the number of "yes" votes given by the crowd in response to the prompt, "*Select all commands that are similar to this one.*" The more "yes" votes, the more similar the crowd felt two symbols were. In grouping like symbols, the challenge is to determine how many *distinct* symbols emerged for a given referent.

This challenge amounts to a graph clustering problem where, for a single referent, the symbols are nodes in a fully-connected graph with weighted edges between all nodes being the ratio of "yes" similarity votes to total votes (Figure 2). The clustering problem is to form sets of nodes such that nodes *within* the same set have maximal similarity while nodes *across* different sets have minimal similarity. This problem is a version of the correlation clustering problem for general weighted graphs, which is of class APX-HARD [8].

We wrote various optimization algorithms for this problem: hill climbing, shotgun hill climbing, simulated annealing, a genetic algorithm, and correlation clustering. We briefly describe the formulation of each algorithm and then present results of testing the algorithms on our crowd-supplied data.

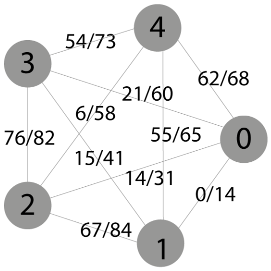


Figure 2. A small example of a fully-connected vote-weighted graph with five nodes. Weights are visible at the midpoints of the edges between nodes. The challenge of clustering is evident with nodes 2, 3, and 4, as nodes 2 & 3 and 3 & 4 have strong affinity, but 2 & 4 do not. So should {2,3,4} be grouped? The same problem exists for nodes 0, 1, and 4, with 0 & 4 and 1 & 4 having strong affinity, but 0 & 1 do not. So should {0,1,4} be grouped? Our actual data had 43 nodes per referent, not just 5.

Hill Climbing

We implemented steepest-ascent hill climbing [30] (pp. 111-112) as a baseline optimization algorithm. A solution state was represented as a set-of-sets, with each subset containing the nodes (symbols) deemed similar. Thus, if all symbols were deemed similar, the set-of-sets would contain one set of all nodes. If no symbols were deemed similar, the set-of-sets would contain a separate set each with only one node.

A fitness function for the hill climber's state was defined as follows. In the set-of-sets, for each pair of symbols *within* a set, if the "yes" votes (out of all votes) passed a one-sided binomial test,[4] we added the percentage of "yes" votes to the fitness score. If, on the other hand, the binomial test failed, we subtracted one minus the percentage of "yes" votes from the fitness score. Conversely, for each pair of symbols *across* different sets, we did the opposite: passing the binomial test subtracted the percentage of "yes" votes from the fitness score, while failing the binomial test added one minus the percentage of "yes" votes to the fitness score. Defined in this way, the fitness score drove the climber towards states having more similar symbols grouped together and less similar symbols grouped separately.

An essential component to any hill climber is formulating the possible moves to neighboring states. For our climber in a given state represented by a set-of-sets, neighboring states were created by taking each symbol and placing it in every other set, including the empty set. Thus, for a state with N sets containing a total of M items, $N \times M$ possible neighboring states were considered by the climber at each move. As a steepest-ascent hill climber, the move yielding the greatest fitness gain was always chosen. A concern with this approach, of course, is getting stuck on local maxima within the search space.

Shotgun Hill Climbing

We also implemented a variant of hill climbing called shotgun hill climbing [30] (pp. 112-113), which is equivalent to random-restart hill climbing. In shotgun hill climbing, multiple climbers are "shot" randomly into the search space and climb from wherever they land, thereby increasing the chances of at least one climber reaching the global maximum. We first utilized 1000 climbers but surprisingly found that five climbers performed just as well (and much faster), suggesting a search space with few local maxima. Our shotgun hill climber therefore used five climbers.

Simulated Annealing

A more sophisticated iterative improvement algorithm than hill climbing is simulated annealing [30] (pp. 113-114). In our implementation, a random move was chosen from among neighboring states. If that move was to a better state, it was always taken. If it was not, then unlike hill climbing, it still *might* have been taken depending on a probability that starts high and decreases over time, like the temperature of a cooling metal. Following Kirkpatrick *et al.* [18], we automatically set the initial and minimum temperatures via stochastic sampling of the search space. We set the cooling rate to 3% and enabled 100 possible moves per temperature.

[4] Our one-sided binomial test examined whether there was a statistically significant proportion of "yes" votes out of all votes indicating similarity between two symbols at the $\alpha = .05$ level.

Genetic Algorithm

Genetic algorithms have been used to form clusters within graphs [39]. We implemented a genetic algorithm [29] that began with 1000 randomly generated "organisms," each encoding a set-of-sets solution state, and evolved them for up to 10,000 generations, mutating offspring by moving a random symbol into a random set, including the empty set. Offspring mutated on a decreasing schedule, with more mutations in early generations than in later generations. At each generation, 20% of the fittest organisms survived to populate the next generation. An exception rate of 5% was used to retain less-fit organisms for subsequent generations to avoid local maxima.

Correlation Clustering

Unlike general iterative improvement algorithms, correlation clustering is specific to the problem of clustering nodes in graphs that have "affinity," while avoiding clustering nodes that have "aversion." Correlation clustering was first formalized by Bansal *et al.* [3] for graphs with binary weighted edges represented as $\langle +, - \rangle$. Ailon *et al.* [2] introduced a correlation clustering approximation algorithm for general weighted graphs in which each edge had a positive weight w^+ and a negative weight w^-. However, our problem is different, as edges are not negatively weighted; all are positively weighted, some more than others. We implemented the KWIKCLUSTER approximation algorithm of Ailon *et al.* [1] (p. 23:13) but defined "affinity" as passing the one-sided binomial test described above, *i.e.*, node pairs that did not pass this test were deemed to have "aversion." As this algorithm is highly dependent on the selection of random nodes, we ran it with 1000 random restarts, taking the best outcome.

Performance Results

To assess the performance of each algorithm in grouping our symbols, we performed a simple experiment. We first used trial-and-error to find the settings for each algorithm that seemed to perform best within reasonable time limits. We then ran the grouping algorithms on all 43 symbols for each of our 10 referents. We did this 10 times and averaged the results. Thus, we were left with 5 algorithms × 10 referents = 50 fitness scores and execution times (Table 2). Ultimately, we selected shotgun hill climbing for Crowdsensus because of its high fitness scores and fast execution times.

Solution Fitness Scores

Our statistical analysis shows that fitness scores conformed to the assumptions of analysis of variance. A repeated measures ANOVA using the Greenhouse-Geisser correction [11] indicated no significant effect of the algorithms on fitness scores ($F_{1.2,\ 10.7} = 2.32$, *n.s.*).

Algorithm Execution Times

Algorithm execution times violated the normality assumption of ANOVA and were therefore analyzed with nonparametric tests. A Friedman test indicated a significant effect of algorithm on execution time ($\chi^2_{(4,N=50)} = 37.60$, $p < .0001$). *Post hoc* Wilcoxon signed-rank tests corrected with Holm's sequential Bonferroni procedure for multiple comparisons [14] indicated that all execution times were significantly different under this analysis ($p < .02$), except the genetic algorithm and hill climbing.

On balance, given the parity in fitness scores and the desire for fast execution times, it seems everything but simulated annealing is a viable option. As noted, we chose shotgun hill climbing for its peak fitness score and reasonable execution time. It was noteworthy that hill climbing performed so well, indicating that the search space must be relatively smooth. Also, the fast speed of correlation clustering, even with 1000 random restarts, was impressive.

Algorithm	Solution Fitness	Execution Time (sec)
Hill climbing	561.3 (141.8)	14.5 (6.1)
Shotgun hill climbing	562.6 (141.1)	100.3 (47.7)
Simulated annealing	562.6 (141.1)	282.0 (241.8)
Genetic algorithm	560.5 (141.1)	17.1 (1.2)
Correlation clustering	559.9 (144.3)	2.8 (1.8)

Table 2. Means (and standard deviations) of the quality of the solutions produced by the symbol-clustering algorithms, and how long it took to produce them, in seconds. Higher fitness scores indicate better performance. Lower execution times are preferred.

RESULTS: VALIDATING THE CROWDSENSUS APPROACH

We used data from 410 crowd workers who provided votes that passed our validation tests out of 461 workers who began our study. The 51 workers whose data we did not use triggered our spam detectors. We used the shotgun hill climbing algorithm, described above, to cluster symbols based on the crowd's votes. We scrutinize Crowdsensus by:

1. Calculating the **agreement score** for each referent (see Eq. 1). We wanted to understand how much the crowd and the experts each thought the users who elicited the symbols were in agreement.
2. Quantitatively measuring the **similarity of the symbol groupings** produced by the crowd and by the experts.
3. Finding the **definitive symbol** for each referent. We wanted to see what symbols the crowd chose for each referent, which ones the experts' judgments converged on, and what overlap there was between the experts and the crowd.
4. Calculating the **cost** of using Crowdsensus to analyze the symbols, comparing this to the cost of using experts.
5. Calculating the **time** it took the crowd to analyze the symbols, comparing this to the time for experts.

Agreement Scores

We calculated the agreement scores from the crowd and from the experts. Vatavu and Wobbrock [35] recommend using qualitative judgments with agreement scores and provided a guide for these judgements. According to their guide, low agreement scores are less than 0.1, medium scores are between 0.1 – 0.3, high scores are between 0.3 – 0.5, and very high agreement scores are above 0.5.

Figure 3 shows how each of the experts, including the first author, and the crowd (via Crowdsensus) grouped the symbols for each referent. High or very high agreement scores indicated that whoever grouped the symbols thought the end users who provided these symbols exhibited some consensus. The experts tended to generate fewer groupings with larger numbers of symbols leading to high agreement scores, meaning the experts thought there was high similarity among the symbols for most referents. The crowd (plus clustering algorithm, *i.e.*, Crowdsensus) had low agreement scores for five of the 10 referents, meaning the crowd was stricter than experts in its assessment of which symbols were similar to each other. This strictness led the crowd to make small groups of very similar symbols.

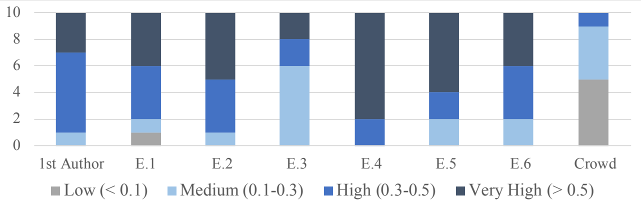


Figure 3. The amount of agreement among symbols elicited for each of the 10 referents as grouped by the first author, the experts (E.1-6), and the crowd via Crowdsensus.

Similarity of Groupings

To determine how similar the various groupings were, we devised a distance metric by which to compare groupings. For each referent, we compared the groupings generated with Crowdsensus to the groupings generated from the experts' judgments by converting each grouping into a 2-D matrix of 43 rows and 43 columns, each row or column representing one symbol. Each cell was populated with one of three values: (-1, 0, +1). A -1 was for unused cells like the intersection of a symbol with itself. A 0 meant the symbols intersecting at this cell belonged to different groups. A +1 meant the two symbols intersecting at this cell belonged to the same group.

An example will help illustrate. Consider the grouping {(0,2), (1)}. Where symbols 0 and 2 belong to the same group and symbol 1 is in a group by itself. We would represent this grouping as shown in Figure 4A. Another grouping of these same symbols could be {(0), (1,2)}, where symbol 0 is in a group by itself, and symbols 1 and 2 are in a group. This second grouping is represented in Figure 4B. To measure the distance between the two groupings, we can use Eq. 2 [16], below:

$$d_2(A,B) = \sqrt{\sum_{i=1}^{n-1}\sum_{j=0}^{i-1}(a_{ij} - b_{ij})^2} \qquad (2)$$

In Eq. 2, the 2-D distance d_2 between 0-based indexed matrices A and B is given by taking the root of the sum of squared differences between all cells a_{ij} and b_{ij} "above the diagonal." A cell a_{ij} is a cell in $n \times n$ matrix A and indicates whether items i and j are in a set together (+1) or not (0). The outer summation causes index i to start in column 1, and the inner summation causes index j to start at row 0. Thus, rows are iterated within columns for cells for which $j < i$.

A

j\i	**0**	**1**	**2**
0	-1	0	+1
1	-1	-1	0
2	-1	-1	-1

B

j\i	**0**	**1**	**2**
0	-1	0	0
1	-1	-1	+1
2	-1	-1	-1

Figure 4. (A) Matrix *A* represents the grouping {(0,2), (1)}. **(B)** Matrix *B* represents the grouping {(0), (1,2)}. This type of matrix representation allows us to compute the distance between two sets-of-sets. Note that the matrices have 0-based indices and assume that their items are indexed likewise from zero.

Using Eq. 2, the distance d_2 between matrices (A, B) in Figure 4 is 0.67, which makes intuitive sense because 1/3 of the elements in A must be moved to create B (*i.e.*, move the 2). In general, then, the 2-D distance d_2 between two matrices is a value from 0, if they are identical, to 1, if the two matrices are entirely different.

Table 3 (next page) shows the mean pairwise distances (and standard deviations) among Crowdsensus, the first author, and each one of the experts averaged over 10 referents. The crowd's groupings via Crowdsensus are similar to the experts', with E.3 being the closest on average to the crowd's groupings at 0.27 and E.4 the furthest at 0.59. The average distance for all the experts among themselves was 0.29.

Definitive Symbol Selection

The third measure we used to judge the crowd's performance relative to the experts was to find the definitive symbol generated by the crowd workers, *i.e.*, the largest group of similar symbols, and compare that to the definitive symbol that emerged from the experts for each referent. We used an "experts' group" to achieve our goal. The experts' group for each referent was made up of symbols that appeared across all of the largest symbol groups per referent generated by the first author and the experts.

Figure 5 (next page) shows the number of symbols in the experts' group for every referent, the number of symbols the crowd elected, and the number of overlapping symbols between the two. For referents 2, 3, and 10, all of the symbols selected by the crowd appeared in the experts' group, with the crowd's symbols making up 73%, 42%, and 56% of the experts' group, respectively. For referents 1, 5, 8, and 9, the crowd's symbols made up 81%, 100%, 30%, and 59% of the experts' group, respectively.

Referent 4 was a case where there was no overlap in symbols between the crowd and the experts, *i.e.*, Crowdsensus converged on a different set of symbols than the experts' merged judgements to invoke "switch between pages in a backward direction." For referents 6 and 7, we were unable to find a single referent that all experts had in their individual definitive symbol groups, which resulted in empty experts' groups for these two referents. The crowd's votes generated

	Crowd	1st Author	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
Crowd	0	.39 (.14)	.40 (.23)	.41 (.17)	.27 (.18)	.59 (.14)	.41 (.21)	.39 (.19)
1st Author		0	.39 (.14)	.36 (.10)	.20 (.15)	.40 (.12)	.39 (.10)	.37 (.10)
Expert 1			0	.20 (.10)	.36 (.14)	.27 (.17)	.18 (.11)	.14 (.11)
Expert 2				0	.37 (.90)	.29 (.16)	.19 (.07)	.15 (.08)
Expert 3					0	.48 (.13)	.36 (.15)	.36 (.12)
Expert 4						0	.24 (.15)	.26 (.18)
Expert 5							0	.18 (.10)
Expert 6								0

Table 3. The mean distances over 10 referents between Crowdsensus, the first author, and the experts. The distance values are in [0.0, 1.0], where 0.0 means the two matrices are identical, and 1.0 means they are entirely different. Standard deviations are in parentheses.

a definitive symbol for referent 6, but the crowd's votes did not cluster any two symbols into a group to be used as the definitive symbol for referent 7. Table 4 presents the experts' and crowd's sets of definitive symbols *(i.e.*, elicited voice commands).

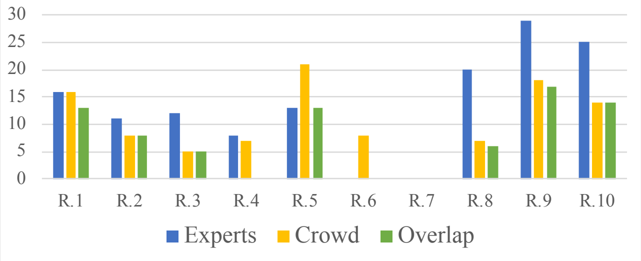


Figure 5. Comparison of symbol counts in the definitive group for each referent (R1-R10). Blue bars are the number of symbols in the experts' group. Yellow bars are for the crowd. Green bars are the number of overlapping symbols in the experts' and crowd's groups.

Referent *How would you phrase a voice command that would…*	**Experts' symbol**	**Crowd's symbol**
1. open the browser?	"Open browser"	"Open browser"
2. perform a search for cat videos?	"Search for cat videos"	"Search for cat videos"
3. click the first link on the page?	"Click the first link"	"Click the first link"
4. switch between pages in a backward direction?	"Go back 'one / to the last / to the previous' page"	"Go to the previous page"
5. switch between pages in a forward direction?	"Go to the next page"	"Go to the next page"
6. switch between multiple open tabs?	N/A	"Switch tabs"
7. select a region on the page?	N/A	N/A
8. request the same page you are browsing to be loaded again?	"Reload the page"	"Reload the page"
9. bookmark a page?	"Bookmark this page"	"Bookmark this page"
10. close the browser?	"Close browser"	"Close browser"

Table 4. A list of all referents used in this study (adapted from [23]), and the symbols that would invoke them, as chosen by experts and non-expert crowd workers. An "N/A" indicates a lack of convergence.

Cost

We paid each researcher $50 to analyze our dataset of 10 referents. Therefore, the cost of one referent to be analyzed by one expert was $5.00. For this study, with its 43 symbols per referent, we needed 2.15 Mechanical Turk HITs to complete one referent because one HIT contained 20 List Comparison voting tasks (see Figure 1B), and we needed 43 such tasks to compare all of the symbols to each other. Our Turk tasks were priced at $2.75. Thus, the cost for a crowd worker to analyze a referent was $5.90. In summary, then, the cost of Crowdsensus was similar to the cost of the experts.

Time

On average, it took the experts, including the first author, 3.17 (*SD*=1.2) minutes to group the symbols for a single referent. For the crowd workers, one HIT consisted of 20 List Comparison tasks (see Figure 1B). HITs were batched into groups of 31. Therefore, a batch resulted in 31 × 20 = 620 List Comparison tasks. To analyze a single referent, we needed 43 List Comparison tasks. In 11.22 minutes, the crowd analyzed 14.4 referents, meaning, on average, it took about 0.78 minutes per referent. This result was about four times faster than the experts (*i.e.*, 3.17 / 0.78 = 4.06).

DISCUSSION

To understand our findings in context, we revisit the research questions posed in the introduction:

***Q1.** How can the crowd facilitate similarity judgments for agreement analysis in end-user elicitation studies?*

We were able to successfully utilize *Crowdsensus* to produce agreement analyses for our study. Crowdsensus takes a dataset of user-generated symbols and creates custom web interfaces. These interfaces facilitate the gathering of similarity judgment votes. Clustering algorithms group symbols during the agreement analysis of elicitation studies.

We pilot-tested potential interfaces for Crowdsensus, examining the possible effects of simple user interface choices—displaying 1:1, 1:*N*, or *N*:*N* simultaneous comparisons; the specific phrasing of instructions; and the number of symbols displayed within a single HIT. Iterative comparative testing and refinement of the interface helped us identify a set of interface parameters that produce high-quality crowd judgements, particularly the use of a 1:*N* selection mechanism and the crucial phrase "similar to" in the instructions. We discovered that it is safe to present large symbol sets as smaller, manageable subsets.

From our study, we discovered that the average time a crowd worker spent voting on the similarity of one symbol compared to 43 others was 38.23 seconds. We recommended using 15 seconds as a threshold to accept valid data. We also used an approach to ensure the crowd workers were paying attention by testing whether they voted-as-similar the exact symbol to which they were comparing all other symbols.

Q2. *By using the crowd, what are the benefits, if any, in terms of cost and time compared to the status quo use of experts' judgments?*

In our study, the crowd cost $5.90 per referent, while the experts cost $5.00. Therefore, the crowd cost was similar to the experts. Specific monetary costs may vary in practice depending on the exact wages of experts and crowd workers and on the desired level of redundancy in crowd judgments.

The average time it took the crowd to analyze a single referent was 0.78 minutes. The average time it took the expert researchers we recruited to examine our data was 3.17 minutes per referent. Therefore, the crowd can provide similarity judgments to analyze end-user elicitation studies using our List Comparison interface (see Figure 1B) about four times faster than expert researchers.

Having the crowd analyze the results of elicitation studies cuts time significantly. The researcher is also free to conduct further analysis and tweak the crowd's groupings, capitalizing on her own expertise and whatever supplementary material (*e.g.*, study notes) she has available. By cutting down on the resources needed to analyze elicitation studies, we open up numerous possibilities to advance this methodology, like scaling up the number of symbols collected to create more inclusively-designed technologies, and lowering barriers to conduct replication studies or reanalyze published studies.

Q3. *How does the quality of the results produced by the crowd compare to those produced by expert researchers?*

Similarity judgments for symbols generated in elicitation studies are subjective. We expected to find differences in the way people grouped symbols. We therefore discuss the quality of the crowd's results compared to those of our experts along three lines: agreement scores, grouping similarity, and the definitive symbol for each referent.

Agreement Scores. Overall, the crowd's grouping-agreement scores were lower than those of the experts. The experts seemed to believe that the symbols were more similar to each other than did the crowd. The crowd was stricter than the experts in considering which symbols belonged with each other in the same group, thereby creating smaller sets of more-similar symbols.

Grouping Similarity. The average distances between the crowd's grouping and each one of the experts' was similar to those of the experts among themselves for every referent. This finding means that the crowd produced groupings comparable to those produced by the experts.

Definitive Symbol Selection. Due to the lower agreement scores mentioned above, the crowd's groupings were of smaller similar symbol sets. Thus, a definitive symbol chosen by the crowd contained less variance than those from the experts. For example, for referent 10, both the crowd and experts chose "close browser;" however, the experts' grouping included variants like "please close this browser for me," while the crowds' grouping required a stricter match. This situation is where the experience of the experts and their familiarity with elicitation studies gives them an advantage over the crowd. Experts conducting elicitation studies might choose to create synonyms, like in the case where there are two or more popular groups of symbols proposed for a given referent. In our analysis, we made the simplifying assumption that a designer would choose a single symbol for each referent. However, the output of Crowdsensus could easily be analyzed by experts to form symbol synonyms.

Limitations

The symbols we elicited were text strings, a decision we made to simplify the process of capturing symbols and being able to share them easily with external experts. We suspect that richer multimedia symbols will require more time to analyze; we plan to generalize Crowdsensus to eventually support rich media. Also, our experts worked independently. Typically, if more than one researcher is analyzing the results, they work together to reach consensus on definitive symbols. Also, our dataset was relatively small, at 10 referents and 430 symbols. Some studies are larger, such as Wobbrock *et al.*'s [38], which had 27 referents and 1080 symbols, or Kane *et al.*'s [17], which had 22 referents and 880 symbols. The scope of our dataset was limited to the standard functionality of a web browser, which made it easier to analyze than novel interactions; it is unknown whether expertise beyond that of crowd workers is required in more complex domains.

CONCLUSION

In this work, we developed *Crowdsensus*, a system that extends the popular design method of end-user elicitation studies by allowing researchers to crowdsource the crucial similarity judgments central to agreement analysis. Our work demonstrated that it is possible to use a crowd of non-experts, in conjunction with automatic clustering algorithms, to successfully analyze the results of an elicitation study. We also showed that using the crowd comes with benefits like saving time; it also produces results similar to those obtained with experts. It is our hope that using the crowd can propel end-user elicitation further by lowering barriers to running these studies at scale and with diverse audiences.

ACKNOWLEDGMENTS

This work was supported in part by funding from Microsoft Research, the Mani Charitable Foundation, and the National Science Foundation under grant IIS-1702751. Any opinions, findings, conclusions or recommendations expressed in our work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

1. Ailon, N., Charikar, M. and Newman, A. (2005). Aggregating inconsistent information: Ranking and clustering. *Proceedings of the ACM Symposium on Theory of Computing.* New York: ACM Press, pp. 684-693.
2. Ailon, N., Charikar, M. and Newman, A. (2008). Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM 55* (5), pp. 23:1-23:27.
3. Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering. *Machine Learning 56* (1-3), pp. 89-113.
4. Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D. and Panovich, K. (2010). Soylent: a word processor with a crowd inside. *Proceedings of the ACM symposium on User interface software and technology*. New York: ACM Press, pp. 313-322.
5. Bragg, J. and Weld, D.S. (2013) Crowdsourcing multi-label classification for taxonomy creation. *Proceedings of AAAI Conference on Human Computation and Crowdsourcing.* Palm Springs, CA: HCOMP, pp. 25-33.
6. Chan, J., Dang, S. and Dow, S.P. (2016). Improving crowd innovation with expert facilitation. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work & Social Computing.* New York: ACM Press, pp. 1223-1235.
7. Chilton, L.B., Little, G., Edge, D., Weld, D.S. and Landay, J.A. (2013). Cascade: Crowdsourcing taxonomy creation. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1999-2008.
8. Demaine, E.D., Emanuel, D., Fiat, A. and Immorlica, N. (2006). Correlation clustering in general weighted graphs. *Theoretical Computer Science 361* (2-3), pp. 172-187.
9. Findlater, L., Lee, B. and Wobbrock, J.O. (2012). Beyond QWERTY: Augmenting touch screen keyboards with multi-touch gestures for non-alphanumeric input. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 2679-2682.
10. Good, M.D., Whiteside, J.A., Wixon, D.R. and Jones, S.J. (1984). Building a user-derived interface. *Communications of the ACM 27* (10), pp. 1032-1043.
11. Greenhouse, S.W. and Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika 24* (2), pp. 95-112.
12. Guo, S., Parameswaran, A., and Garcia-Molina, H. (2012). So who won?: dynamic max discovery with the crowd. *Proceedings of the ACM International Conference on Management of Data.* New York: ACM Press, pp. 385-396.
13. Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*. Cham: Springer, pp. 84-92.
14. Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics 6* (2), pp. 65-70.
15. Hou, W., Chen, K., Li, H., Zhou, H. (2018). User defined eye movement-based interaction for virtual reality. *Proceedings of the International Conference on Cross-Cultural Design*. Cham: Springer, pp. 18-30.
16. https://math.stackexchange.com/questions/507742/distance-similarity-between-two-matrices:
17. Kane, S.K., Wobbrock, J.O. and Ladner, R.E. (2011). Usable gestures for blind people: Understanding preference and performance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 413-422.
18. Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science 220* (4598), pp. 671-680.
19. Kittur, A., Smus, B., Khamkar, S. and Kraut, R.E. (2011). Crowdforge: Crowdsourcing complex work. *Proceedings of the ACM Symposium on User interface Software and Technology*. New York: ACM Press, pp. 43-52.
20. Kühnel, C., Westermann, T., Hemmert, F., Kratz, S., Müller, A. and Möller, S. (2011). I'm home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human-Computer Studies 69* (11), pp. 693-704.
21. Morris, M.R., Danielescu, A., Drucker, S., Fisher, D., Lee, B., Schraefel, M.C. and Wobbrock, J.O. (2014). Reducing legacy bias in gesture elicitation studies. *ACM Interactions 21* (3), pp. 40-45.
22. Morris, M.R., Wobbrock, J.O. and Wilson, A.D. (2010). Understanding users' preferences for surface gestures. *Proceedings of Graphics Interface*. Toronto: Canadian Information Processing Society, pp. 261-268.
23. Morris, M.R. (2012). Web on the wall: insights from a multimodal interaction elicitation study. *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York: ACM Press, pp. 95-104.
24. Nebeling, M., Ott, D., & Norrie, M. C. (2015). Kinect Analysis: A system for recording, analysing and sharing multimodal interaction elicitation studies. *Proceedings of SIGCHI Symposium on Engineering Interactive Computing Systems*. New York: ACM Press, pp. 142-151.
25. Obaid, M., Häring, M., Kistler, F., Bühling, R. and André, E. (2012). User-defined body gestures for navigational control of a humanoid robot. *Proceedings of the International Conference on Social Robotics*. Berlin: Springer, pp. 367-377.
26. Piper, A.M., Campbell, R., and Hollan, J.D. (2010). Exploring the accessibility and appeal of surface computing for older adult health care support.

Proceedings of the ACM Conference on Human Factors in Computing Systems. New York: ACM Press, pp. 907-916.

27. Piumsomboon, T., Clark, A., Billinghurst, M. and Cockburn, A. (2013). User-defined gestures for augmented reality. *Proceedings of INTERACT 2013*. Berlin: Springer, pp. 282-299.
28. Vedantam, R., Zitnick, C. L., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. *Proceedings of the IEEE conference on computer vision and pattern recognition.* Pp. 4566-4575.
29. Stuart J. Russell and Peter Norvig. (1995). Genetic algorithms and evolutionary programming. In *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, pp. 619-621.
30. Stuart J. Russell and Peter Norvig. (1995). Iterative improvement algorithms. In *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, pp. 111-114.
31. Silberman, M.S., B. Tomlinson, R. LaPlante, J. Ross, L. Irani, and A. Zaldivar. (2018). Responsible research with crowds. *Communications of the ACM*. 61 (3), pp. 39-41.
32. Speicher, M. and Nebeling, M. (2018). GestureWiz : A human-powered gesture design environment for user interface prototypes. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp.1-11.
33. Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai. A.T., (2011). Adaptively learning the crowd kernel. *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, USA, pp. 673-680.
34. Vatavu, R.D. (2012). User-defined gestures for free-hand TV control. *Proceedings of the 10th European Conference on Interactive TV and Video*. New York: ACM Press, pp. 45-48.
35. Vatavu, R.D. and Wobbrock, J.O. (2016). Between-subjects elicitation studies: Formalization and tool support. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 3390-3402.
36. Vatavu, R.D. and Wobbrock, J.O. (2015). Formalizing agreement analysis for elicitation studies: New measures, significance test, and toolkit. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1325-1334.
37. Wobbrock, J.O., Aung, H.H., Rothrock, B. and Myers, B.A. (2005). Maximizing the guessability of symbolic input. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1869-1872.
38. Wobbrock, J.O., Morris, M.R. and Wilson, A.D. (2009). User-defined gestures for surface computing. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1083-1092.
39. Zhang, Z., Cheng, H., Chen, W., Zhang, S. and Fang, Q. (2008). Correlation clustering based on genetic algorithm for documents clustering. *Proceedings of the IEEE Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, pp. 3193-3198.