Human Computation (2019) 6:1:42-55

ISSN: 2330-8001, DOI: 10.15346/hc.v6i1.3

Autocompletion interfaces make crowd workers slower, but their use promotes response diversity

XIPEI LIU, UNIVERSITY OF VERMONT

JAMES P. BAGROW, UNIVERSITY OF VERMONT

ABSTRACT

Creative tasks such as ideation or question proposal are powerful applications of crowdsourcing, yet the quantity of workers available for addressing practical problems is often insufficient. To enable scalable crowdsourcing thus requires gaining all possible efficiency and information from available workers. One option for text-focused tasks is to allow assistive technology, such as an autocompletion user interface (AUI), to help workers input text responses. But support for the efficacy of AUIs is mixed. Here we designed and conducted a randomized experiment where workers were asked to provide short text responses to given questions. Our experimental goal was to determine if an AUI helps workers respond more quickly and with improved consistency by mitigating typos and misspellings. Surprisingly, we found that neither occurred: workers assigned to the AUI treatment were slower than those assigned to the non-AUI control and their responses were more diverse, not less, than those of the control. Both the lexical and semantic diversities of responses were higher, with the latter measured using word2vec. A crowdsourcer interested in worker speed may want to avoid using an AUI, but using an AUI to boost response diversity may be valuable to crowdsourcers interested in receiving as much novel information from workers as possible.

1. INTRODUCTION

Crowdsourcing applications vary from basic, self-contained tasks such as image recognition or labeling (Welinder and Perona, 2010) all the way to open-ended and creative endeavors such as collaborative writing, creative question proposal, or more general ideation (Little et al., 2010). Yet scaling the crowd to very large sets of creative tasks may require prohibitive numbers of workers. Scalability is one of the key challenges in crowdsourcing: how to best apply the valuable but limited resources provided by crowd workers and how to help workers be as efficient as possible. Efficiency gains can be achieved either collectively at the level of the entire crowd or by helping individual workers. At the crowd level, efficiency can be gained by assigning tasks to workers in the best order (Tran-Thanh et al., 2013), by filtering out poor tasks or workers, or by best incentivizing workers (Allahbakhsh et al., 2013). At the individual worker level, efficiency gains can come from

helping workers craft more accurate responses and complete tasks in less time.

One way to make workers individually more efficient is to computationally augment their task interface with useful information. For example, an autocompletion user interface (AUI) (Sevenster et al., 2012), such as used on Google's main search page, may speed up workers as they answer questions or propose ideas. However, support for the benefits of AUIs is mixed and existing research has not considered short, repetitive inputs such as those required by many large-scale crowdsourcing problems. More generally, it is not yet clear what are the best approaches or general strategies to achieve efficiency gains for creative crowdsourcing tasks.

In this work, we conducted a randomized trial of the benefits of allowing workers to answer a text-based question with the help of an autocompletion user interface. Our experimental goal was to determine how workers would use an AUI and how an AUI may affect their responses. Would they be faster at answering such short questions by saving on typing time? Or would the cognitive load of reading the AUI as it appeared and updated slow down the worker, even enough to offset any savings from faster text entry? Further, would the AUI lead to more consistent responses across workers by mitigating typos, or less consistent responses, by providing novel suggestions for workers to consider or by acting as a cognitive primer?

In our randomized trial, workers interacted with a web form that recorded how quickly they entered text into the response field and how quickly they submitted their responses after typing is completed. After the experiment concluded, we measured response diversity using textual analyses and response quality using a followup crowdsourcing task with an independent population of workers. Our results indicate that the AUI treatment did not affect quality, and did not help workers perform more quickly nor achieve greater response consensus (including typos). Instead, workers with the AUI were significantly slower and their responses were more diverse than workers in the non-AUI control group.

2. RELATED WORK

An important goal of crowdsourcing research is achieving efficient scalability of the crowd to very large sets of tasks. Efficiency in crowdsourcing manifests both in receiving more effective information per worker and in making individual workers faster and/or more accurate. The former problem is a significant area of interest (Karger et al., 2014; Li et al., 2016; McAndrew et al., 2017) while less work has been put towards the latter.

One approach to helping workers be faster at individual tasks is the application of usability studies. Kittur et al. (2008) famously showed how crowd workers can perform user studies, although this work was focused on using workers as usability testers for other platforms, not on studying crowdsourcing interfaces. More recent usability studies on the efficiency and accuracy of workers include: Cheng et al. (2015), who consider the task completion times of macrotasks and microtasks and find workers given smaller microtasks were slower but achieve higher quality than those given larger macrotasks; Lasecki et al. (2015), who study how the sequence of tasks given to workers and interruptions between tasks may slow workers down; and Maddalena et al. (2016), who study completion times for relevance judgment tasks, and find that imposed time limits can improve relevance quality, but do not focus on ways to speed up workers. These studies do not test the effects of the task interface, however, as we do here.

The usability feature we study here is an autocompletion user interface (AUI). AUIs are broadly familiar to online workers at this point, thanks in particular to their prominence on Google's main search bar (evolving out of the original Google Instant implementation). However, literature on the benefits of AUIs (and related word prediction and completion interfaces) in terms of improving efficiency is decidedly mixed.

It is generally assumed that AUIs make users faster by saving keystrokes (Bast and Weber, 2006). However, there is considerable debate about whether or not such gains are countered by increased cognitive load induced by processing the given autocompletions (Koester and Levine, 1994). Anson et al. (2006) showed that typists can enter text more quickly with word completion and prediction interfaces than without. However, this study focused on a different input modality (an onscreen keyboard) and, more importantly, on a text transcription task: typists were asked to reproduce an existing text, not answer questions. Sevenster et al. (2012) showed that medical typists saved keystrokes when using an autocompletion interface to input standardized medical terms. However, they did not consider the elapsed times required by these users, instead focusing on response times for the AUI suggestions to appear, and so it is unclear if the users were actually faster with the AUI. There is some evidence that long-term use of an AUI can lead to improved speed and not just keystroke savings (Magnuson and Hunnicutt, 2002), but it is not clear how general such learning may be, and whether or not it is relevant to short-duration crowdsourcing tasks.

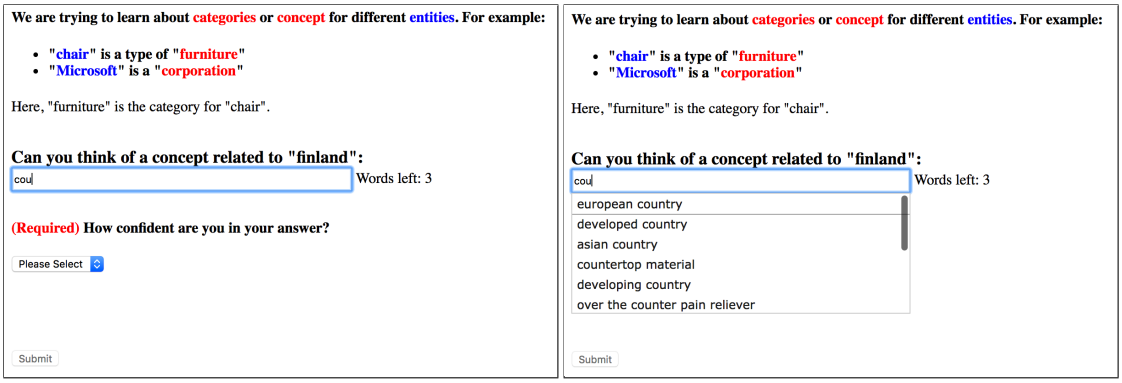
3. EXPERIMENTAL DESIGN

Here we describe the task we studied and its input data, worker recruitment, the design of our experimental treatment and control, the "instrumentation" we used to measure the speeds of workers as they performed our task, and our procedures to post-process and rate the worker responses to our task prior to subsequent analysis.

Task description and question data For this work, we focused on a conceptualization or "IsA" task. Each task consisted of a question of the form: "*FOO* is a type of:" followed by a short one-line text field for the worker to respond. The particular terms "*FOO*" then defines each question. Before this question was a brief description of the task followed by two examples: "*chair* is a type of furniture" and "*Microsoft* is a corporation". See Fig. 1.

The question terms ("chair" and "Microsoft" in the above examples) were chosen from the Microsoft Concept Graph (MCG) dataset (Wu et al., 2012; Wang et al., 2015). These data provide a bipartite knowledge graph linking *entities* to *concepts*, for example "city" is a concept related to the entity "Berlin". We chose these data for our conceptualization task so that we have a comparative baseline, as the MCG captures the same relationships we measure in our task.

We chose 10 entities randomly from the MCG to act as question terms. The MCG data are somewhat noisy, heavily skewed to rare terms (often medical terms), and contain many abstract entity–concept relations, so we first performed a filtering step to focus on commonplace and easy-to-understand question terms. We also required that 5 of the chosen terms be one-word entities longer than two letters and 5 be multi-word phrases, both without numbers. See Table 1 for our final chosen question terms.



(A) Control form *(B) Autocompletion User Interface (AUI) form*

***Figure 1.** Screenshots of our conceptualization task interface. The presence of the AUI is the only difference between the task interfaces.*

ID	*Question term*	*ID*	*Question term*
Q1	hail	Q6	occupational therapist
Q2	millet	Q7	standard deviation
Q3	steam	Q8	motor vehicle
Q4	finland	Q9	dengue fever
Q5	spider	Q10	citric acid

***Table 1.** Question terms used in our conceptualization task. Workers were shown these questions in random order.*

Crowdsourcing and treatment We recruited workers on Amazon Mechanical Turk (AMT) to perform our task. Recruited workers must have 80% or better approval rating, be USA-located, and be able to view adult content. Each human intelligence task (HIT) was one conceptualization task, i.e. one of the ten questions. Workers could perform anywhere from one to ten HITs. Questions were shown to each worker in random order. Each worker response generates a question-response text pair which may or may not be unique as other workers may give the same response to the same question. Workers were compensated $0.05 per HIT.

Workers were blindly assigned to one of two conditions with equal probability (simple random assignment) when they accepted their first HIT. This assignment was then carried over for any subsequent HITs performed by that worker. The control group consisted of a HIT interface (web form) with a text entry field without an autocompletion user interface (AUI). We refer to this as the Control form and the workers assigned to the Control form as the Control group. The treatment consisted of a text entry field but with an associated AUI; corresponding to the Control group, we refer to this form as the AUI form and the workers assigned to the AUI form as the AUI group. Screenshots comparing Control and AUI forms are shown in Fig. 1.

In all other respects the HIT interfaces were identical. In particular, for both forms, JavaScript was

used on the field to prevent workers from inputting punctuation or responses exceeding four words. Copy or paste is prevented on the page; workers can only fill in the text entry by typing or, if it is available, by selecting from the AUI. The HIT was not submittable until the response field was filled.

Autocompletion user interface The AUI we used was implemented with jQuery-UI's (ver. 1.12.1) autocomplete widget with autofocus enabled[1]. Whenever two or more characters are present in the response field, a search based on the current contents of the response field is triggered of a database containing all MCG concepts with at least 5 associated entities ($n = 705{,}710$ concept terms). The MCG dataset is very exhaustive and tries to list as many associations between entities and concepts as possible; our filtering criteria removes the large number of domain-specific, often medical, jargon. Further, due to its automatic collection, the MCG concepts may potentially contain typos (see also our typo analysis in Sec. 4.3). Concept terms are indexed for speed and the search term is matched from both sides using MySQL's "LIKE" operator, and the first six matches are dynamically displayed in the AUI (Fig. 1B) with up to another six available by scrolling. The LIKE operator also provided the ordering of suggestions in the AUI. The search repeats whenever the current response changes; the AUI disappears if there are fewer than two characters present in the response field. Workers were not required to select a response from the AUI. An AUI will only be as useful as the data it queries, and will likely not affect worker responses if it does not provide relevant suggestions. Thus we chose the MCG concepts for the AUI to offer meaningful autocompletions for our conceptualization task.

Instrumentation To study the effects of the AUI, each HIT form was instrumented with JavaScript to record the times when workers first entered text into the response field, when they last entered text into the response field, and when the form was submitted. Note that while we also recorded the time when the HIT was accepted, we did not use these data because it is unclear when a worker accepts a HIT as opposed to when a worker actually begins work on that HIT (AMT workers sometimes open a series of HITs into separate browser tabs, and then later process those HITs). Due to this, our future experiments will also record when the browser window containing the HIT is "focused"[2].

This instrumentation allows us to measure two important features of worker activity:

i. Typing duration—Total elapsed time between the first and last keypress made by the worker into the text area.

ii. Submission delay—Total elapsed time between the final keypress into the text area and the submission of the form.

Response processing and quality ratings Worker responses were post-processed by removing casing and transforming any whitespace to a single space character. Additional processing was unnecessary because of the in-browser processing done by the form (see above).

[1] Autofocus makes it easy for the worker to quickly select the top AUI response.

[2] Although it is still possible that the window may be focused but the worker is not actively using the HIT, combining focus logging with key press and mouse activity logging should give a reasonable signal for when the worker is interacting with the HIT and when she is not.

A second, non-experimental set of HITs was given to workers to measure the perceived quality of each unique question-response pair. Instead of using additional workers to rate responses, the quality of responses for our conceptualization task could be assessed computationally using, for example, ontology datasets. However, combining free text responses from workers with a fixed-vocabulary dataset is a challenging natural language processing task beyond the scope of this work, so here we simply relied on ratings by independent workers. Evaluation workers were shown the same instructions and examples as conceptualization workers and then were shown statements of the form "Q: Can you think of a concept related to *FOO*? A: *BAR*", where *BAR* is a given worker response to question term *FOO*. These workers were asked to rate their agreement with this statement on a 1–5 rating scale (1—least agree; 5—strongest agree). Each worker was shown ten such statements per HIT, randomly sampled from both Control and AUI responses, and was compensated at a rate of $0.25 per HIT. Evaluation workers had to meet the same selection requirements as conceptualization workers (80% approval rating, etc.) Workers who belong to either Control or AUI groups were excluded from these tasks.

4. RESULTS

4.1. Data collection

We recruited 176 AMT workers to participate in our conceptualization task. Of these workers, 90 were randomly assigned to the Control group and 86 to the AUI group. These workers completed 1001 tasks: 496 tasks in the control and 505 in the AUI. All responses were gathered within a single 24-hour period during April 2017.

After Control and AUI workers were finished responding, we initiated our non-experimental quality ratings task. Whenever multiple workers provided the same response to a given question, we only sought ratings for that single unique question and response. Each unique question-response pair ($n = 428$) was rated at least 8–10 times (a few pairs were rated more often; we retained those extra ratings). We recruited 119 AMT workers (who were not members of the Control or AUI groups) who provided 4300 total ratings.

4.2. Differences in response time

We found that workers were slower overall with the AUI than without the AUI. In Fig. 2 we show the distributions of typing duration and submission delay. There was a slight difference in typing duration between Control and AUI (median 1.97s for Control compared with median 2.69s for AUI)[3]. However, there was a strong difference in the distributions of submission delay, with AUI workers taking longer to submit than Control workers (median submission delay of 7.27s vs. 4.44s). This is likely due to the time required to mentally process and select from the AUI options. We anticipated that the submission delay may be counter-balanced by the time saved entering text, but the total typing duration plus submission delay was still significantly longer for AUI than control (median 7.64s for Control vs. 12.14s for AUI). We conclude that the AUI makes workers significantly slower.

We anticipated that workers may learn over the course of multiple tasks. For example, the first time a worker sees the AUI will present a very different cognitive load than the 10th time. This learning

[3]Responses from the AUI group were slightly longer than those from the Control; median length of 11 characters vs. 9 characters.

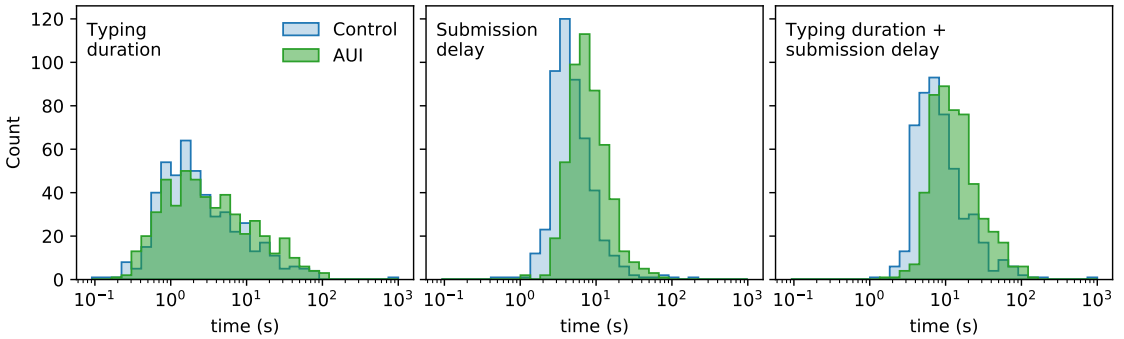


Figure 2. ***Distributions of time delays. Workers in the AUI treatment were significantly slower than in the control, and this was primarily due to the submission delay between when they finished entering text and when they submitted their response.***

may eventually lead to improved response times and so an AUI that may not be useful the first time may lead to performance gains as workers become more experienced.

To investigate learning effects, we recorded for each worker's question-response pair how many questions that worker had already answered, and examined the distributions of typing duration and submission delay conditioned on the number of previously answered questions (Fig. 3). Indeed, learning did occur: the submission delay (but not typing duration) decreased as workers responded to more questions. However, this did not translate to gains in overall performance between Control and AUI workers as learning occurred for both groups: Among AUI workers who answered 10 questions, the median submission delay on the 10th question was 8.02s, whereas for Control workers who answered 10 questions, the median delay on the 10th question was only 4.178s. This difference between Control and AUI submission delays was significant (Mann-Whitney test: $U = 872$, $n_{\text{Control}} = 61$, $n_{\text{AUI}} = 53$, $p < 10^{-4}$). In comparison, AUI (Control) workers answering their first question had a median submission delay of 10.97s (7.00s). This difference was also significant (Mann-Whitney test: $U = 9822$, $n_{\text{Control}} = 169$, $n_{\text{AUI}} = 165$, $p < 10^{-5}$). We conclude that experience with the AUI will not eventually lead to faster responses those of the control.

4.3. Differences in response diversity

We were also interested in determining whether or not the worker responses were more consistent or more diverse due to the AUI. Response consistency for natural language data is important when a crowdsourcer wishes to pool or aggregate a set of worker responses. We anticipated that the AUI would lead to greater consistency by, among other effects, decreasing the rates of typos and misspellings. At the same time, however, the AUI could lead to more diversity due to providing better suggestions than those a worker could provide on their own or even due to cognitive priming: seeing suggested responses from the AUI may prompt the worker to revise their initial response.

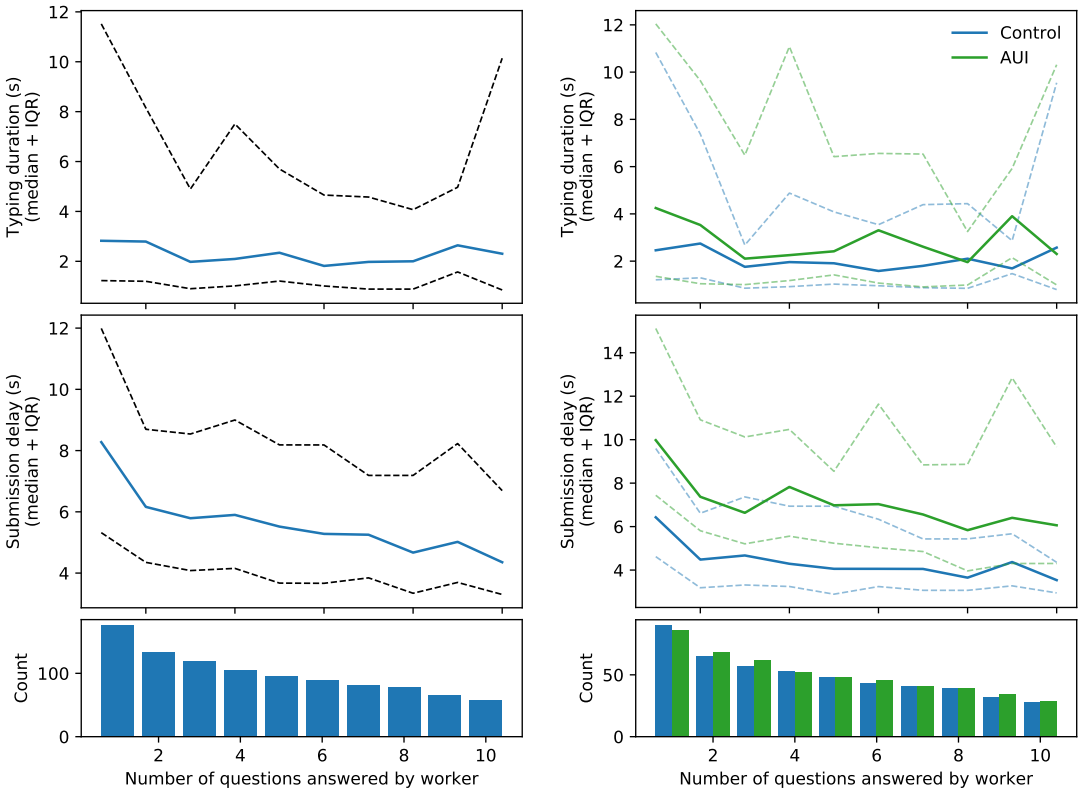


Figure 3. ***Workers became faster as they gained experience by answering more questions, but this improvement occurred in both Control and AUI groups.***

Increased diversity[4] may be desirable when a crowdsourcer wants to receive as much information as possible from a given task.

To study the lexical and semantic diversities of responses, we performed three analyses. First, we aggregated all worker responses to a particular question into a single list corresponding to that question. Across all questions, we found that the number of unique responses was higher for the AUI than for the Control (Fig. 4A), implying higher diversity for AUI than for Control.

Second, we compared the diversity of individual responses between Control and AUI for each question. To measure diversity for a question, we computed the number of responses divided by the number of unique responses to that question. We call this the *response density*. A set of responses has a response density of 1 when every response is unique but when every response is the same, the

[4]We focus on diversity across workers, whether or not they give the same or similar responses. Individual responses could be diverse in that they possess many meanings (polysemy), but if all workers give the same (polysemous) response, then the set of responses would still have low diversity, particularly lexical diversity.

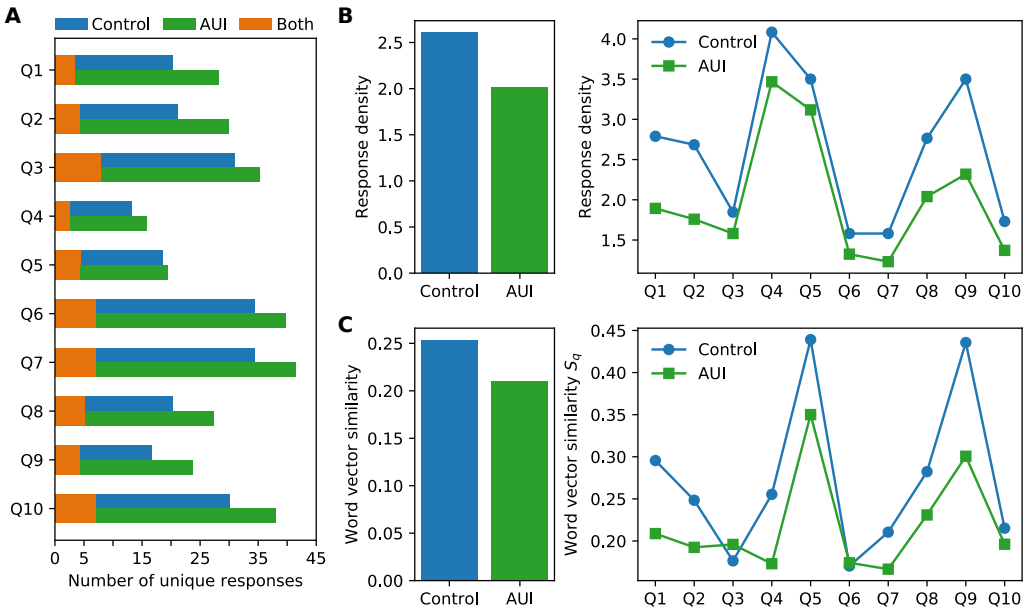


Figure 4. ***AUI workers had more lexically (A, B) and semantically (C) diverse responses than Control workers.***

response density is equal to the number of responses. Across the ten questions, response density was significantly lower for AUI than for Control (Wilcoxon signed rank test paired on questions: $T = 0, n = 10, p < 0.01$) (Fig. 4B).

Third, we estimated the semantic diversity of responses using word vectors. Word vectors, or word embeddings, are a state-of-the-art computational linguistics tool that incorporate the semantic meanings of words and phrases by learning vector representations that are embedded into a high-dimensional vector space (Mikolov et al., 2013a,b). Vector operations within this space such as addition and subtraction are capable of representing meaning and interrelationships between words (Mikolov et al., 2013b). For example, the vector $\mathbf{v}_{\text{king}} + \mathbf{v}_{\text{woman}} - \mathbf{v}_{\text{man}}$ is very close to the vector $\mathbf{v}_{\text{queen}}$, indicating that these vectors capture analogy relations. Here we used 300-dimension word vectors trained on a 100B-word corpus taken from Google News[5] (word2vec). For each question we computed the average similarity between words in the responses to that question—a lower similarity implies more semantically diverse answers. Specifically, for a given question q, we concatenated all responses to that question into a single document D_q, and averaged the vector similarities $sim(\mathbf{v}_i, \mathbf{v}_j)$ of all pairs of words $(w_i, w_j), w_i \neq w_j$ in D_q, where $\mathbf{v}_i$ is the word vector

[5]https://code.google.com/archive/p/word2vec

corresponding to word w_i:

$$S_q \equiv \frac{\sum_{i=1}^{|D_q|-1} \sum_{j=i+1}^{|D_q|} sim(\mathbf{v}_i, \mathbf{v}_j)\,(1-\delta_{ij})}{\sum_{i=1}^{|D_q|-1} \sum_{j=i+1}^{|D_q|} (1-\delta_{ij})}, \tag{1}$$

where $\delta_{ij} = 1$ if $w_i = w_j$ and zero otherwise. We also excluded from (1) any word pairs where one or both words were not present in the pre-trained word vectors (approximately 13% of word pairs). For similarity $sim(\mathbf{v}_i, \mathbf{v}_j)$ we chose the standard *cosine similarity* between two vectors. As with response density, we found that most questions had lower word vector similarity S_q (and are thus collectively more semantically diverse) when considering AUI responses as the document D_q than when D_q came from the Control workers (Fig. 4C). The difference was significant (Wilcoxon signed rank test paired on questions: $T = 4$, $n = 10$, $p < 0.05$).

Taken together, we conclude from these three analyses that the AUI suggestions increased the diversity of the responses workers gave.

Typo analysis Related to response diversity is the extent of typos found in worker responses. An AUI may reduce the number of typos, and reducing typos will help with entity recognition, disambiguation and record linkage tasks a crowdsourcer may perform in order to process and collate a body of responses. To measure typos we applied the Enchant spellchecker library[6] to the text generated by workers using Enchant's builtin American English library. Casing is important for spelling proper nouns—'european' is misspelled while 'European' is not—but our text is recorded as lowercase only. To account for this, we simply consider any incorrectly spelled word as spelled correctly if its capitalized form (capitalizing the first letter) is considered correctly spelled.

Applying Enchant to the worker response text we see that the overall rate of typos is quite low. Specifically, we found that control workers generated a total of 692 words, of which 12 were detected as containing typos. Meanwhile, AUI workers generated a total of 900 words, of which 14 were detected as containing typos. The rate of typos for control workers was 1.73% while for AUI workers it was 1.56%, a small difference that was not significant (one-sided proportions test: $z = 0.278, p = 0.39$).

This typo analysis confirms that the AUI does lower the rate of typos, but the rate was not significantly different between the control and AUI responses.

4.4. No difference in response quality

Following the collection of responses from the Control and AUI groups, separate AMT workers were asked to rate the quality of the original responses (see Experimental design). These ratings followed a 1–5 scale from lowest to highest. We present these ratings in Fig. 5. While there was variation in overall quality across different questions (Fig. 5A), we did not observe a consistent difference in perceived response quality between the two groups. There was also no statistical difference in the overall distributions of ratings per question (Fig. 5B). We conclude that the AUI neither increased nor decreased response quality.

[6] https://abiword.github.io/enchant/

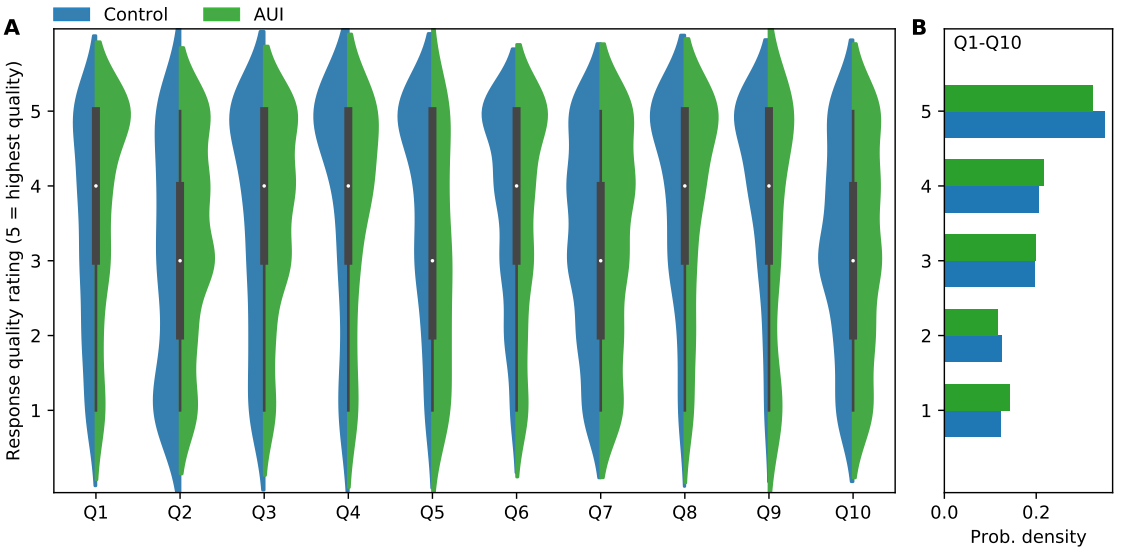


Figure 5. ***Quality of responses. All question-response pairs were rated independently by workers on a 1-5 scale of perceived quality (1–lowest quality, 5–highest quality).***

5. DISCUSSION

We have showed via a randomized control trial that an autocompletion user interface (AUI) is not helpful in making workers more efficient. Further, the AUI led to a more lexically and semantically diverse set of text responses to a given task than if the AUI was not present. The AUI also had no noticeable impact, positive or negative, on response quality, as independently measured by other workers, or on the rate of typos.

Autocompletion user interfaces are very popular, appearing as part of web browser address bars, search engine interfaces, and programmer text editors or integrated development environments. Therefore, it seems reasonable that crowdsourcers may want to use an AUI when building a crowdsourcing interface for a text-oriented task, especially since the popularity of AUIs makes it likely most crowd workers will understand their use. Indeed, we planned to build a crowdsourcing interface with an AUI for a particular task. However, after finding how mixed the research was on the efficacy of AUIs, we decided to conduct the simplified timing experiment reported here to see if the AUI saves total time or only typing time, and we found, at least within the scope of our experiment, that it does not save total time. It seems likely that other researchers or designers of crowdsourcing tasks may also wish to augment their own task interfaces with AUIs, making our results here valuable to other crowdsourcing researchers.

A challenge with text-focused crowdsourcing is aggregation of natural language responses. Unlike binary labeling tasks, for example, normalizing text data can be challenging. Should casing be removed? Should words be stemmed? What to do with punctuation? Should typos be fixed? One of our goals when testing the effects of the AUI was to see if it helps with this normalization task,

so that crowdsourcers can spend less time aggregating responses. We found that the AUI would likely not help with this in the sense that the sets of responses became more diverse, not less. Yet, this may in fact be desirable—if a crowdsourcer wants as much diverse information from workers as possible, then showing them dynamic AUI suggestions may provide a cognitive priming mechanism to inspire workers to consider responses which otherwise would not have occurred to them.

One potential explanation for the increased submission delay among AUI workers is an excessive number of options presented by the AUI. The goal of an AUI is to present the best options at the *top* of the drop down menu (Fig. 1B). Then a worker can quickly start typing and choose the best option with a single keystroke or mouse click. However, if the best option appears farther down the menu, then the worker must commit more time to scan and process the AUI suggestions. Our AUI always presented six suggestions, with another six available by scrolling, and our experiment did not vary these numbers. Yet the size of the AUI and where options land may play significant roles in submission delay, especially if significant numbers of selections come from AUI positions far from the input area.

We aimed to explore position effects, but due to some technical issues we did not record the positions in the AUI that workers chose. However, our Javascript instrumentation logged worker keystrokes as they typed so we can approximately *reconstruct* the AUI position of the worker's ultimate response. To do this, we first identified the logged text inputed by the worker before it was replaced by the AUI selection, then used this text to replicate the database query underlying the AUI, and lastly determined where the worker's final response appeared in the query results. This procedure is only an approximation because our instrumentation would occasionally fail to log some keystrokes and because a worker could potentially type out the entire response even if it also appeared in the AUI (which the worker may not have even noticed). Nevertheless, most AUI workers submitted responses that appeared in the AUI (Fig. 6A) and, of those responses, most were found in the first few (reconstructed) positions near the top of the AUI (Fig. 6B). Specifically, we found that 59.3% of responses were found in the first two reconstructed positions, and 91.2% were in the first six. With the caveats of this analysis in mind, which we hope to address in future experiments, these results provide some evidence that the AUI responses were meaningful and that the AUI workers were delayed by the AUI even though most chosen responses came from the top area of the AUI which is most quickly accessible to the worker.

Beyond AUI position effects and the number of options shown in the AUI, there are many aspects of the interplay between workers and the AUI to be further explored. We limited workers to performing no more than ten tasks, but will an AUI eventually lead to efficiency gains beyond that level of experience? Likewise, an AUI may be more effective for different types of tasks than the text-oriented crowdsourcing task we investigated here, for example searching an address book or writing computer code. Lastly, an AUI is only as useful as the suggestions it delivers, and this depends on both the quality of the dataset it queries and the algorithm used to rank those queries. It is therefore possible an AUI will lead to efficiency gains when applying more advanced autocompletion and ranking algorithms than the one we used. Given that workers were slower with the AUI primarily due to a delay after they finished typing which far exceeded the delays of non-AUI workers, better algorithms may play a significant role in speeding up or, in this case, slowing down workers. Taken together, our results here indicate that crowdsourcers must be very judicious and consider many potential factors when deciding whether or not to augment crowd workers with autocompletion user

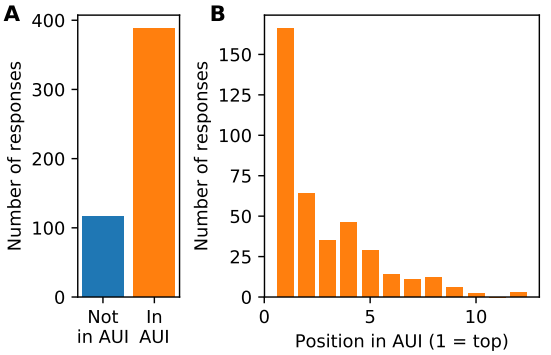


Figure 6. Inferred positions of AUI selections based on the last text workers in the AUI group typed before choosing from the AUI. (A) Most submitted AUI responses appeared in the AUI. (B) Among the responses appearing in the AUI, the reconstructed positions of those responses tended to be at the top of the AUI, in the most prominent, accessible area.

interfaces.

ACKNOWLEDGMENTS

We thank S. Lehman, J. Bongard, and the anonymous reviewers for useful comments and gratefully acknowledge the resources provided by the Vermont Advanced Computing Core. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1447634.

6. REFERENCES

Allahbakhsh, M, Benatallah, B, Ignjatovic, A, Motahari-Nezhad, H. R, Bertino, E, and Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing* 17, 2 (2013), 76–81.

Anson, D, Moist, P, Przywara, M, Wells, H, Saylor, H, and Maxime, H. (2006). The effects of word completion and word prediction on typing rates using on-screen keyboards. *Assistive technology* 18, 2 (2006), 146–154.

Bast, H and Weber, I. (2006). Type less, find more: fast autocompletion search with a succinct index. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 364–371.

Cheng, J, Teevan, J, Iqbal, S. T, and Bernstein, M. S. (2015). Break It Down: A Comparison of Macro- and Microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 4061–4064.

Karger, D. R, Oh, S, and Shah, D. (2014). Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research* 62, 1 (2014), 1–24.

Kittur, A, Chi, E. H, and Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, 453–456.

Koester, H. H and Levine, S. P. (1994). Modeling the speed of text entry with a word prediction interface. *IEEE Transactions on Rehabilitation Engineering* 2, 3 (1994), 177–187.

Lasecki, W. S, Rzeszotarski, J. M, Marcus, A, and Bigham, J. P. (2015). The Effects of Sequence and Delay on Crowd Work. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 1375–1378.

Li, Q, Ma, F, Gao, J, Su, L, and Quinn, C. J. (2016). Crowdsourcing High Quality Labels with a Tight Budget. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. ACM, 237–246.

Little, G, Chilton, L. B, Goldman, M, and Miller, R. C. (2010). Exploring Iterative and Parallel Human Computation Processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10)*. ACM, 68–76.

Maddalena, E, Basaldella, M, De Nart, D, Degl'Innocenti, D, Mizzaro, S, and Demartini, G. (2016). Crowdsourcing relevance assessments: The unexpected benefits of limiting the time to judge. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

Magnuson, T and Hunnicutt, S. (2002). Measuring the effectiveness of word prediction: The advantage of long-term use. *TMH-QPSR* 43, 1 (2002), 57–67.

McAndrew, T. C, Guseva, E, and Bagrow, J. P. (2017). Reply & Supply: Efficient crowdsourcing when workers do more than answer questions. *PLOS ONE* 12, 8 (2017), e69829.

Mikolov, T, Chen, K, Corrado, G, and Dean, J. (2013)a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

Mikolov, T, Sutskever, I, Chen, K, Corrado, G. S, and Dean, J. (2013)b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119.

Sevenster, M, van Ommering, R, and Qian, Y. (2012). Algorithmic and user study of an autocompletion algorithm on a large medical vocabulary. *Journal of Biomedical Informatics* 45, 1 (2012), 107–119.

Tran-Thanh, L, Venanzi, M, Rogers, A, and Jennings, N. R. (2013). Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 901–908.

Wang, Z, Wang, H, Wen, J.-R, and Xiao, Y. (2015). An Inference Approach to Basic Level of Categorization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 653–662.

Welinder, P and Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 25–32.

Wu, W, Li, H, Wang, H, and Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 481–492.