

The Knowledge Accelerator: Big Picture Thinking in Small Pieces

Nathan Hahn¹, Joseph Chang¹, Ji Eun Kim², Aniket Kittur¹

Carnegie Mellon University¹
Pittsburgh, PA USA

{nhahn, josephcc, nkittur}@cs.cmu.edu

Bosch Corporate Research²
Pittsburgh, PA USA

JiEun.Kim@us.bosch.com

ABSTRACT

Crowdsourcing offers a powerful new paradigm for online work. However, real world tasks are often interdependent, requiring a big picture view of the difference pieces involved. Existing crowdsourcing approaches that support such tasks — ranging from Wikipedia to flash teams — are bottlenecked by relying on a small number of individuals to maintain the big picture. In this paper, we explore the idea that a computational system can scaffold an emerging interdependent, big picture view entirely through the small contributions of individuals, each of whom sees only a part of the whole. To investigate the viability, strengths, and weaknesses of this approach we instantiate the idea in a prototype system for accomplishing distributed information synthesis and evaluate its output across a variety of topics. We also contribute a set of design patterns that may be informative for other systems aimed at supporting big picture thinking in small pieces.

Author Keywords

information synthesis; crowdsourcing; crowd work; complex workflow; design patterns.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Crowdsourcing is a powerful mechanism for accomplishing work online. By decomposing and distributing the cognitive work of an individual, crowdsourcing can provide a larger pool of resources more quickly and with lower transaction costs than through traditional work. A common emerging theme is that the more a task can be split, simplified, and distributed into smaller subtasks, and the lower the cost of accepting and completing a task, the larger the pool of workers accessible who can complete it anywhere at anytime [36]. For example, microtask markets such as Amazon Mechanical Turk (AMT) enable hundreds of thousands of workers from across the globe to be recruited within seconds [8].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3362-7/16/05...\$15.00

<http://dx.doi.org/10.1145/2858036.2858364>

However, much work in the real world is not amenable to crowdsourcing because of the difficulty in decomposing tasks into small, independent units. As noted by many researchers [9, 38, 47, 48], decomposing tasks ranging from writing an article to creating an animated film often results in pieces that have complex dependencies on each other. Take for example the goal of writing an article that synthesizes information on the web about a given topic (e.g., growing better tomatoes). Coming up with a coherent and comprehensive set of topics (e.g., soil, sunlight, watering, pruning) is challenging without a global view of the data. The need for coherence extends throughout the fractal nature of the article: each section, paragraph, and sentence must have a proper transition and flow. Supporting such work requires having a big picture view of different pieces at different scales and ensuring they all fit together.

Accomplishing big picture thinking through small tasks is challenging because it means that each person can only have a limited view of the bigger picture. As a result, many of the applications of crowdsourcing have been limited to simple tasks such as image labeling where each piece can be decomposed and processed independently. Those approaches that do crowdsource tasks requiring big picture thinking — such as volunteer communities such as Wikipedia, open source software, or paid crowd work approaches such as flash teams [59] or Turkomatic [41] — have relied on a heavily invested contributor such as a moderator or an experienced contributor to maintain the big picture. For example, in Wikipedia a large proportion of the work is done by a small group of heavily invested editors [39], and the quality of an article is critically dependent on there being a small number of core editors who create and maintain a big picture structure for more peripheral members to contribute effectively [35].

A reliance on a single or a small number of individuals to maintain the big picture creates a bottleneck on the size and complexity of task amenable to crowdsourcing, and also results in brittleness: if the person maintaining the big picture leaves, it can cause serious problems for the group task. This is a real problem that online production communities are facing; for example, Wikipedia has identified as a key challenge that it is losing core editors faster than it can attract and grow new ones [62]. As these core editors are disproportionately responsible for not only producing content but also for creating a structure for peripheral contributions, their departure is particularly difficult to handle. Taking a step towards enabling the production of complex artifacts through many con-

tributors making small contributions might thus have implications in reducing individual bottlenecks in microtask markets and beyond.

Our main contribution in this paper is the idea that a computational system can scaffold an emerging interdependent, big picture view entirely through small contributions of individuals, each of whom sees only a part of the whole. To investigate this idea we instantiate it in a working software system to explore the viability, strengths, and weaknesses of the approach, and evaluate the output of the system across a variety of topics. Finally, we also contribute a set of design patterns that may be informative for other systems aimed at supporting big picture thinking in small packages.

System Overview

The “Knowledge Accelerator” (KA) is a prototype system which uses crowd workers each contributing small amounts of effort to synthesize online information for complex and/or open-ended questions. The KA system starts with a given question (such as “How do I deal with the arthritis in my knee as a 28 year old”) and crowdsources the generation of a coherent article that synthesizes different sources, viewpoints, and topics found online relevant to answering the question. Critically, the KA system accomplishes this process without a core overseer or moderator.

As the goal of the system was to probe how to accomplish a complex information synthesis task entirely through relatively small contributions, we limited our maximum task payment to \$1 US, aimed at incentivizing a target task time of approximately 5-10 minutes. We chose this approach because a fixed payment amount matches the structure of many micro-task crowdsourcing markets (e.g., versus a fixed time period of 10 minutes). While some crowdsourcing markets (such as UpWork or eLance) do support hourly rates and fixed time periods, the double-sided transaction (or “handshake”) costs in which employers and workers vet each other in such markets would constitute a substantial fraction of the working time we target, and the time scale of projects in such markets (typically measured in hours) do not match well with the time scale of the projects we target here (i.e., minutes).¹

An example of the output of the system for the target question “How do I get my tomato plants to produce more tomatoes?” can be found in Figure 1. To produce this output workers find high value sources from the web (e.g., gardening.about.com), extract the useful and relevant clips of information from them, cluster these clips across sources into commonly discussed topics (e.g., feeding or pruning), and generate an article for each topic that synthesizes the relevant clips into coherent chunks of information while reducing redundancies (e.g., if several sources all mention soil pH range, the article should not include that information multiple times). Workers also

¹One concern could be that \$1 could motivate different amounts of effort across different countries. For all tasks other than sourcing and clipping we limited the pool of workers for our tasks to U.S. workers to control for cross-country currency differences. For sourcing and clipping workers U.S. workers spent an average of 9.72 minutes and 6.89 minutes respectively, while non-U.S. workers spent 8.65 minutes and 8.36, which were not significantly different.

find relevant multimedia images and video to illustrate each chunk. Information sources used in the creation of the article are referenced in the final output, and the article is organized by subtopics with the most diverse set of references first (See Figure 2 for the KA process overview).

Our primary contribution in addressing this task is to further our theoretical understanding of the mechanisms and limitations of accomplishing big picture thinking in small pieces, which may have implications for crowdsourcing systems that aim to do complex cognitive tasks including micro-task crowdsourcing [36], peer production communities [35], friendsourcing [10], and selfsourcing [63]. However, addressing this task may also have intrinsic utility in paving the road for crowdsourced systems that can synthesize complex information from a variety of sources on demand. Such systems may be especially useful for topics not covered by traditional online sources; examples include low frequency or highly personalized search queries (such as looking for information on a particular medical condition given the person’s context including age or other symptoms), topics whose sources are highly unstructured and distributed (such as advice giving on discussion forums), or for information that is inside an organization’s firewall (such as for a company’s IT support sessions).

Below we discuss the challenges involved in developing the system, particularly focusing on issues central to supporting big picture thinking with workers each seeing only a small part of the whole. We first discuss related work, describe the system architecture, then evaluate the utility of the systems output versus top online sources across a variety of topics.

RELATED WORK

Crowdwork Complex Cognition and Workflow

While most crowdsourcing approaches have focused on simple and/or independent tasks, there is a growing interest in crowdsourcing tasks that tap into complex and higher-order cognition [36]. Many of these fall into the class of decomposing cognitive processing in a structured way such that many workers can contribute [2, 9, 12, 32, 38, 34, 42, 44, 45, 47]. Our work builds on this foundation by incorporating adaptive crowd workflows (e.g., TurKit, JabberWocky, CrowdWeaver), crowd-driven task generation (e.g. CrowdForge, Turkomatic), combining the outputs from decomposed tasks to create a global understanding (e.g., Cascade, Crowd Synthesis) and multi-stage crowd quality control process in which crowds can both generate new versions of output as well as vote on it (e.g., CrowdForge, Soylent, TurKit). However, we go beyond previous work in aiming to support a coherent big picture view while avoiding individual bottlenecks. Doing this is significantly more challenging than the tasks decomposed in prior research, requiring a search for structure during the sampling process, a reliance on novices to function with more context than they enter the task with, and a tight interdependence between each subtask such that any failures could negatively impact the value of the entire artifact.

How Do I Get My Tomato Plants To Produce More Tomatoes?

Contents

1. Tomatos - Feeding
2. Pruning Is Love
3. Maintenance And Harvesting
4. Tomatos - Proper Potting Procedure
5. Weather And Sunlight Conditions
6. Growing Tomatoes
7. Tomatos - Stakes And Support

Tomatos - Feeding

Producing better tomato plants is as simple as picking the perfect soil. There are many market soils or one can add a few things to their own soil. Extra nutrients go a long way in producing more tomatoes per plant.

Tomatoes are heavy feeders since they are smaller plants that depend on the bushy growth to support fruit production. They can benefit from some added nutrition even if you use the best soil. Cutting back on nitrogen will ensure a big, gorgeous pile of fruit coming your way in no time!

Tomatoes take up nutrients the best when the soil pH ranges from 6.2 to 6.8. They need a constant supply of major and minor plant nutrients. Following the rates on the fertilizer label, mix a balanced timed-release or organic fertilizer to the soil as you prepare planting holes.

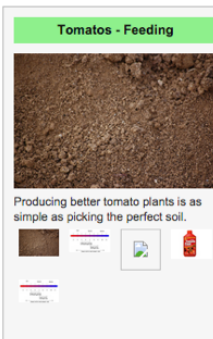
Feeding tomatoes regularly is critical for a good yield. At the very least, you need a good liquid food that is high in potassium.

Any tomato feed from a garden center should do the job. If you want take it a step further, check out Sea Nymph's natural seaweed-based feed or BioBizz's BioGrow, which include molasses to feed the microbes in the soil. About half way through the season, I add a 1 inch (2.5 cm) layer of worm compost or local farm manure to the top of my containers. This adds extra nutrients and soil life.

Amend your plant beds with your own or purchased compost; dry, timed-release fertilizer; and most importantly, worm castings. Add 5 cubic feet of Gardner & Bloome compost; 5 quarts of Gardner & Bloome 4-6-3 Tomato, Herb & Vegetable fertilizer; and a quart of 100% pure worm castings for every 50 square feet of garden space.

References:

- Vertical veg man: how to grow tomatoes successfully (www.theguardian.com)
- Tomatoes..How To Get The Most From Your Plants in The Garden (oldworldgardenfarms.com)
- Love Apple Farms (www.growbetterveggies.com)
- 10 Tips for Growing Great Tomatoes (gardening.about.com)



Producing better tomato plants is as simple as picking the perfect soil.

Figure 1. The final output of the Knowledge Accelerator system.

Information Synthesis

Individual information synthesis is commonly associated with the process of sensemaking. Sensemaking can be characterized as the iterative process of building up a representation of an information space that is useful for achieving the users goal [61]. Theories of sensemaking provide a framework for characterizing and addressing the challenges faced by individuals and can point out leverage points for augmenting the process [61, 20, 40, 65, 24, 18, 53, 58]. Generally, models agree that sensemaking is a dynamic and iterative process involving searching for information; filtering that information based on a user's goals and context; inducing a schema or structure from the information; and applying the schema to take action (e.g., writing a report, making a presentation).

A number of systems have been developed aimed at supporting these stages of sensemaking for an individual user [6, 21, 22, 50, 55, 46] or a group of users working together [35, 39, 54, 56, 57, 65]. However, prior research has focused almost exclusively on situations of integrated sensemaking in which individuals (even in groups) are heavily engaged in the entire sensemaking process. Instead, we aim to distribute the information synthesis process across many different individuals, each of whom may see only a limited view of the process.

Computational approaches to parts of the information synthesis process have also been investigated by many researchers. For example, Question Answering (QA) research addresses the methods and systems that automatically answering questions posted by human in natural language. The complex, interactive QA (ciQA) has been introduced at TREC 2006 and 2007 in addition to factoid and list QA [19]. However, automated QA approaches (and their crowd-based variants [11]) focuses on answering short, factual questions instead

of the complex sensemaking processes we are interested in, where users build up rich mental landscapes of information. Another approach is multi-document summarization [7, 25, 49, 51], which aims to use computational techniques to extract of information from multiple texts written for the same topic using feature based [26], cluster based [28], graph based [23] and knowledge based methods [27]. However, such approaches have limitations in dealing with complex yet short and sparse data that encountered on the web, and do not yet engage in the complex synthesis humans perform, which results in the cohesive and coherent output.

SYSTEM ARCHITECTURE

Broadly, there are two hard problems involved in crowdsourcing information synthesis: learning a good structure for the article based on sampling information from different online sources, and developing a coherent digest given that structure. Below we discuss how the system addresses each of these problems in turn.

Inducing Structure

How can a crowd learn a good structure for an article on an arbitrary topic? Previous crowd approaches such as CrowdForge or CrowdWeaver [38, 34] required workers to decide on a structure up before collecting information on each of these topics. However, these approaches fail when the structure must be learned from the data. For example, few workers will know what the subtopics should be for fixing a Playstations blinking light or for dealing with arthritis; instead, the appropriate structure should emerge from the data. A single individual making sense of a topic often engages in an iterative process of sampling data and building a structure; however, to reduce the latency of having multiple cycles we explore an alternate approach in which the crowd samples a

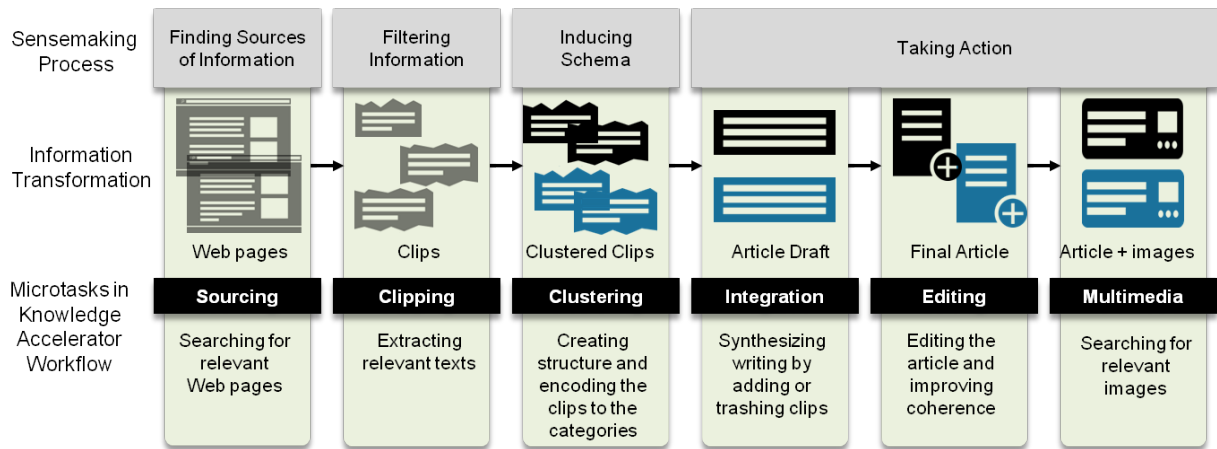


Figure 2. The process of the Knowledge Accelerator (KA), from start to finish

large amount of data in parallel, then leverage a novel hybrid crowd-machine approach that clusters information into topics without requiring any one worker to see the whole picture.

Finding Sources

To search for and filter high quality information sources we asked five workers to each provide the top five web pages relevant to the target question. We found these numbers to work well in practice; future work using optimization approaches [31] could potentially set these dynamically. To ensure high quality responses, for each source we asked workers to report the search term they used and provide a small text clip as “evidence” showing why the source is helpful. This approach appeared to be successful in encouraging workers to find high quality sources: workers made on average 2 different queries ($\sigma = 0.3$), and their more commonly cited sources covered more categories of the structure with fewer sources than choosing sources using standard information retrieval approaches (i.e., using the MMR diversity-based re-ranking algorithm to reorder the sources gathered from the crowdworkers [13]). Sources cited by at least two workers were sent to the filtering stage.

Filtering Information

To filter relevant information snippets from each source workers were presented with one web page and asked to highlight and save at least five pieces of information that would be helpful for answering the question using an interface similar to that described in [37] (Figure 3). One challenge we encountered was that each page could contain a variable amount of useful information, with some long pages having more snippets than a single worker would extract. To spread out worker coverage on long pages, we showed workers sections that had been highlighted by previous workers and asked them to first look for unhighlighted areas when choosing clips. This preference for novelty and surfacing prior workers’ effort allowed us to engage multiple workers for tasks with an unknown amount of relevant information in a more efficient way than simply letting loose many independent workers who would overly focus on the beginning of the page, or having some workers start at the beginning and others at the end [9]. To

focus more effort on potentially rich sources the system dispatches two workers to each source with an additional two workers for every two additional citations a source received.



Figure 3. Workers extract 5 different pieces of relevant information from pages and give it a label

Initially we had workers provide labels to categorize each clip, which we planned to use to develop a structure for the article. However, the lack of context of the bigger picture made these labels poorly suited for inducing a good structure. For example, in Figure 4 the top box shows the category structure induced from labels generated during clipping, while the middle and bottom boxes show the structure induced from the subsequent clustering phase and from a gold standard developed by two independent annotators with access to all clips and sources, respectively. Categories induced from the clipping labels poorly match the gold standard, and include categories with very different abstraction levels (e.g., *Use Drano Max Gel vs tips*). This motivated the development of the subsequent clustering phase.

Clustering

Inducing categories in unstructured collections of text typically requires understanding the global context in order to identify categories that are representative of the information

<p>categories induced during clipping: Boil Water, use hot water, Plunger, try a snake, How to Remove drain stopper, bleach, Use Drano Max Gel, baking soda, drain, tips to unclog, problem, tools, research, internet research, ..., etc.</p>
<p>categories induced after clipping: Hot Water, Plunge, Plunger, Snake the Drain, Remove the Drain Cover, Drain Cleaner, Remove Hair Clusters.</p>
<p>annotator categories: Hot Water, Plunger, Plumbing Snake, Remove Cover, Chemicals, Bent Wire Hanger, Call a Plumber, Shop Vacuum.</p>

Figure 4. Categories induced from different stages for Q1: *How do I unclog my bathtub drain?*

distribution and at appropriate levels of abstraction. The problem of inducing structure without any single worker having a full global context is a particularly challenging problem, and although we describe a basic solution to the problem here for reasons of space and scope, we present a more sophisticated distributed approach in [5] that further generalizes the problem to other domains.

Our approach takes advantage of the fact that many real world datasets have long-tailed distributions, where a few categories make up the bulk of the head of the distribution and many categories with few instances make up the tail. The intuition behind our approach is that first, the crowd can act as a guide to identify the large categories in the head of the distribution, with their judgments training a classifier to categorize the easy cases with high confidence. After automated classification, the crowd can again be used for “clean up”, covering the low-confidence edge cases in the tail of the distribution. This also has the added benefit of easily breaking up the larger question context into sub-contexts for easier consumption in the later parts of the system.

In the first phase, we use workers to label a number of representative categories and leverage those labels to identify meaningful features for an automated classifier. One critical challenge is that workers need to obtain a sense of the distribution of the data without seeing it all. To accomplish this we developed a design we call open-ended set sampling in which workers are presented with four random clips as seeds, and are asked to replace them repeatedly with another random clip until they can determine that the four seed clips belong to meaningfully different categories. Therefore, not only do they have to read the information present in the initial seed clips, but they also need to sample multiple times to understand what “different topics” mean for this dataset. In doing so they are randomly shown new clips, which means they are more likely to encounter categories with probability matching the distribution of topics in the data (i.e., higher probability of encountering larger categories).

After workers pick the seeds, we ask them to highlight discriminative keywords in each of the seed clips which are used to query for similar clips from the full dataset, which the workers then label as *similar* or *different*. With the keyword highlights and the labels created by the workers, we use

an SVM classifier and hierarchical clustering to cluster the high confidence portion of the dataset, sending the uncertain instances to Phase 2.

In the second phase, we employ crowdworkers to clean up the output of the classifier, by presenting them the existing clusters on the left of the screen, and the remaining clips on the right. The workers are first familiarized with the clusters by asking them to review the clips in each cluster and give it a short description. They then categorize the remaining clips into existing clusters or create new clusters if no existing cluster is relevant. These categorization judgments are used to refine the hierarchical clustering model.

Developing a Coherent Article

In this section we describe a set of processes which take as input a set of topics and clips for each topic and output a coherent Wikipedia-like article. There are two core challenges in doing this: first, creating coherence within a topic (e.g., consolidating redundant information); and second, creating coherence between topics (e.g., maintaining consistency across sections).

Integration

Within a single topic, there may be many clips which all contain substantively identical information (e.g., the ideal pH level of soil for growing tomatoes); one goal is to reduce this redundancy so that the final article only describes this information once. At the same time, we recognize the value to seeing that multiple sources all say the same thing; thus, we would like to keep track of all the sources that mention a particular chunk of information. Furthermore, tracking source provenance allows the user to drill back to the original information source in case it is described inaccurately or in a biased way.

To accomplish this we developed an interface in which workers were presented with 5 random clips of information for a given subtopic and asked to integrate that information into a shared text pad. Specifically, they were asked to write the gist of the clip in their own words and transfer the provenance of the clip as a footnote. Missing footnotes triggered a verification check.

Initially, we just instructed individuals to cluster similar items together and insert only the footnote for redundant information. However, we noticed that workers were reluctant to change what they perceived as another workers contributions, consistent with the social blocking found in Andre et al. [4]. This developed into a larger challenge: How could we get workers to gain an understanding of what was in the existing shared pad and feel comfortable modifying it? We introduced a technique we call evaluate then act that requires individuals to read what others have already put into the integrated answer before they are allowed to make a decision about the clip. Our final interface prompts workers to provide specific line numbers corresponding to existing information relevant to their clip, or to explicitly mark their clip as new information or trash. Compared to a version of the system without this structure, significantly more clips were inserted

into the middle of the pad to align better to their given section (13% more, $t(24) = 2.568$, $p < 0.05$) or excluded (11% more, $t(24) = 4.592$, $p < 0.01$) when workers were asked to evaluate before acting.



Figure 5. Editing users the ‘vote-then-edit’ pattern to promote consistency and motivate workers

Editing

We also noticed that coherence needed to be managed not only within topics, but between topics as well. A number of between topic inconsistencies became apparent during the development process, ranging from formatting to structuring to prose. For example, some topics would be organized with bullet points versus paragraphs, and some in the second person point of view versus third person. Previous crowdsourcing approaches have trouble dealing with cross-topic consistency because reading even a single topic can take significant time, let alone reading and editing across all topics. For example, CrowdForge’s [38] approach simply concatenates topics into an article without any attempt at maintaining global coherence. This approach can succeed if the topics and structure either do not require consistency or if they are extremely well specified beforehand: in CrowdForge and CrowdWeaver defining a science article template with clear sections such as *what is the problem*, *what the researchers did*, accomplishes this effectively in a similar manner to core editors specifying a structure in Wikipedia that peripheral members then fill in [35]. However, in the general case such well-defined and pre-specified templates are not always available.

To address this we introduced a new pattern which we call vote-then-edit (Figure 5). This pattern asks workers to first review and vote on and choose the “best” version of a subtopic created by previous workers, while simultaneously getting a sense for commonalities in style, grammatical choices, and organization. They proceed to edit a new subtopic (phase one) or improve on the item they voted on (phase two). In the second case, we expected workers would more carefully select the best version to reduce their future workload, as well as be more motivated to fix issues in it because they had a choice in what they wanted to do.

We used the vote-then-edit pattern in an interleaved “horizontal” and “vertical” workflow. The horizontal phase uses the refined and edited versions of a subtopic section as a “model” for improving the rough output from the integration phase for another subtopic section. Specifically, three workers vote on which of three versions of an edited subtopic

section is the best and then edit a different subtopic subsection using their answer from voting as a model. Their resulting edited output is sent to the vertical phase, in which three workers vote on which of those versions is the best, and are then asked to further improve this now with all of the other subtopic paragraphs presented to them, to ensure the current subtopic has good flow with the other sections. The output from these workers is used in a new horizontal phase, and the cycle continues. The intuition here is that the horizontal phase provides only a single section as a model since there is substantive editing work remaining that requires relatively limited context, while the vertical phase provides all sections because the primary editing work remaining is ensuring consistency across sections. Splitting editing into two interleaved phases with different context-work tradeoffs appeared to be more effective than an older editing approach with a single phase. When we compared the evaluation ratings for the older editing to the interleaved vote-then-edit approach for two questions (Q1 and Q2 in Table 1 respectively), the newer answers were found to be significantly more understandable ($\bar{x} = 0.457$, $p < 0.01$) and helpful ($\bar{x} = 0.373$, $p < 0.05$), suggesting this design pattern helped to create more coherent output.

Multimedia

Images and video can help the reader skim and digest information quickly, as well as provide rich information such as diagrams, instructions, and how-to examples. In our system we enable multimedia from diverse sources to be tied to information blocks, which we define as sections of text demarcated by footnotes. Informally, information blocks correspond to units of information, such as steps in a how-to, or statements or evidence. This has the benefit of ensuring that the images found are specific to pieces of information found in the answer, rather than just being general to the subtopic. For the version of KA described here we did not employ redundancy or voting in the multimedia stage as we did not encounter quality issues; however, since multimedia enrichment is not a particularly interdependent task existing known quality control approaches such as redundancy and voting [36] would likely be sufficient for a production system.

DESIGN PATTERNS

As mentioned in the above task descriptions, during our iterations on each stage we ended up introducing several design patterns that improved the output. Each phase had its own distinctive challenges, yet they still suffered from some of the core challenges highlighted by previous work: motivation, quality-control, and context [36]. Our design patterns served to guide our final system design and add to the set of crowd patterns introduced by previous research [38, 9, 36, 47, 12, 43, 42]. They may be particularly relevant for challenges involving complex interdependent tasks requiring global context for workers seeing only local views.

Context before Action

One of the biggest challenges in crowdsourcing a complex, interdependent task such as information synthesis is providing workers with sufficient global context to perform well despite them having only a local view. Previous researchers

have suggested a variety of useful patterns related to this goal, including making the cost of spurious answers as high as valid ones [33], identifying and surfacing specific sub-task dependencies [41, 59], unified worker interfaces [67] and re-representing tasks in simplified forms [3, 34]. We contribute a set of patterns adding to this literature, specifically focusing on a key tradeoff: given a limited amount of time and effort for an individual worker, how can we provide workers with global context (i.e., investing in their ability to make better decisions) but also engage them in actual production work? Too much invested time providing context reduces the amount of time available for improved task performance.

Open-ended Set Sampling. One challenge with large datasets is giving workers a sense of the distribution of the data despite their observing only subsets of it. This pattern involves a comparison task in which workers are asked to sample random items from the data in order to create a set of non-matching items, as seen in the first step of clustering. A key design factor in this pattern is having a good set function that provides a driver for open-ended sampling and also a stopping point (e.g., when a worker’s familiarity with the distribution gives them a sense that their four seeds represent substantively different topics in the dataset).

Evaluate then Act. In order to get workers to understand the context provided to them, we designed evaluation mechanisms at the beginning of their main task that would allow them to get acquainted with the output from previous workers. This helped workers understand how previous workers processed the information provided to them, improving consistency of the output on parallel tasks, and reducing repeated information. This pattern was leveraged in a number of tasks: clustering, integration, and editing. In the integration phase, we additionally used the evaluation phase to signal to workers that removing others’ work was acceptable and expected, showing that it could be useful in socializing workers into desired procedural practices as well as providing them with context.

Tasks of Least Resistance: Leveraging Worker Choice

Since workers were mostly dealing with dense textual information on a topic they were likely unfamiliar with, we wanted to ensure they were sufficiently motivated. Therefore, we developed a pattern that doubled as both a quality control measure, we well as an incentive for workers. The “task of least resistance” pattern requires that the same crowd worker be involved in two stages of the task, a first stage in which they choose what to work on from a number of alternatives (e.g voting) and a second stage in which they themselves benefit from their choice in terms of having to do less work, easier work, or being able to submit a higher quality output. The intuition is that to minimize their later work workers will choose a foundation that requires the least amount of work possible; i.e., they will choose the “task of least resistance”. This act of choosing is intended to also provide workers with a sense of agency and purpose, which has been shown to increase task performance [15, 60]. This choice also has the potential to increase task performance through workers trying to avoid cognitive dissonance: since workers have themselves

presumably chosen the best quality work to start, poor quality final output could reflect on their own worth [64]. This has a trade off of potentially making tasks longer, more complicated, and more expensive, however the benefit is a higher quality output.

IMPLEMENTATION

The main portion of the application was built using Ruby on Rails and integrated with Amazon’s Mechanical Turk through the Turkee ruby gem [29]. The Ruby on Rails application served as the primary user interface for both the question asker, crowd worker, as well as the answer viewer. A question posed to the system would start the workflow, beginning with source finding. For each stage, after a certain set of conditions were met (number of sources, clips, completed clustering, etc.), the next task in the workflow was automatically started. This allowed the system to run through the entire process with minimal intervention.

The clipping task utilized Readability’s parser API to simplify the appearance of the sources provided during the sourcing phase. This allowed workers to view a cleaner interface in which to clip from, and it also removed some technical limitations involved with clipping from pages that might be multi-paged (readability combines these into one long document) or featured heavy javascript functionality that would interfere with the clipper tool.

For the first phase of the structure induction tasks, the TfIdf-Similarity ruby gem is used for searching clips similar to the seed clips [52]. LIBSVM is used for combining the crowd judgments and cluster a large portion of the dataset [16]. For the integration and editing tasks, we utilized the Etherpad-lite text pad library [1] to allow workers to simultaneously work on the same output.

EVALUATION

To evaluate the usefulness and coherence of the system’s output we compared it to sources an individual might use if they were to complete this task without the KA system. This would most likely involve the use a search engine such as Google to gather information and use existing information sources to learn about the topic. Therefore, as an evaluation, we had a separate set of crowd workers perform a pairwise comparison of the KA output to that of top results returned by Google and those found useful by multiple crowd workers.

Method

Participants were recruited through the AMT US-only pool and paid \$1.50 for the evaluation task. Each participant was randomly assigned to compare the output from the KA system with an existing top website for that question. An individual could only provide one rating per question, but could do the rating task for more than one question. We removed 34 of the 1385 unique participants who provided an evaluation rating who also participated in a KA system task.

The “top websites” used in the comparison task were the top five Google results, as well as any additional Google results

that were highly cited (mentioned by 3 or more turkers) during the sourcing phase of the system. Some questions had a larger number of highly cited sources, resulting in more additional websites, as can be seen in Figure 6.

In the evaluation task, participants were first asked a series of questions that would cause them to read and understand both sources. In order to encourage quality through defensive task design [33], for the output from the KA system and the existing web page, they were asked to list the different sections on each and three different keywords that would describe those sections. After they read and parsed each web page, they were presented with a brief persona of a friend who was having the problem posed to the KA system. Workers were then asked, for that problem, to rate the comprehensiveness, confidence, helpfulness, trustworthiness, understandability, and writing of each web page on a seven point Likert scale (from 1 to 7) and provide an explanation for their rating on each dimension. We averaged ratings on these dimensions into a single score representing the overall perceived quality of the page.

We selected 11 target questions for evaluation by browsing question and answer forums, Reddit.com, and referencing on-line browsing habits [14]. For some questions, we added some additional constraints to test the performance of the system for more personalized questions. In addition to this external evaluation, we also had the crowdworkers who participated in the KA system fill out a short feedback form detailing their experience using the system. We ask three questions about the difficulty of the task, the clarity of the instructions provided, and the easy of use of the user interface. We recorded some brief demographics about our workers, including to the country they were from.

Results

Aggregating across all questions, KA output was rated significantly higher than the comparison web pages, which included the top 5 Google results and sources cited more than 3 times (KA: $\bar{x} = 2.904$ vs Alt. Sites: $\bar{x} = 2.545$, $t(1493) = 13.062$, $p < 0.001$). An analysis of individual questions corrected for multiple comparisons is shown in Table 1.

The strongly positive results found were surprising because some of the websites in the comparison set were written by experts and had well-established reputations. Only on the two travel questions, Barcelona ($\bar{x} = -0.109$) and LA ($\bar{x} = -0.044$), and the VIM question ($\bar{x} = 0.180$) did the KA output not significantly outperform the comparison pages. A closer examination of these pages suggests that for the two travel questions, because of the strong internet commodity market surrounding travel, a considerable amount of effort has been spent on curating good travel resources. Even with the slightly more specific LA query, there were still two specialized sites dedicated to attraction for kids in LA (Momypoppins.com and ScaryMommy.com). The VIM question represented a mismatch between our output and the question style. A number of the sources for the question were tutorials, however in the clipping phase, these ordered tutorials were broken up into unordered clips, creating an information model breakdown. This points out an interesting limitation in

Question	N	Score
Q1: <i>How do I unclog my bathtub drain?</i>	116	0.292 *
Q2: <i>How do I get my tomato plants to produce more tomatoes?</i>	177	0.420 *
Q3: <i>What are the best attractions in LA if I have two little kids?</i>	158	-0.044
Q4: <i>What are the best day trips possible from Barcelona, Spain?</i>	98	-0.109
Q5: <i>My Worcester CDi Boiler pressure is low. How can I fix it?</i>	139	0.878 *
Q6: <i>2003 Dodge Durango has an OBD-II error code of P440. How do I fix it?</i>	138	0.662 *
Q7: <i>2005 Chevy Silverado has an OBD-II error code of C0327. How do I fix it?</i>	135	0.412 *
Q8: <i>How do I deal with the arthritis in my knee as a 28 year old?</i>	139	0.391 *
Q9: <i>My Playstation 3 has a solid yellow light, how do I fix it?</i>	119	0.380 *
Q10: <i>What are the key arguments for and against Global Warming?</i>	138	0.386 *
Q11: <i>How do I use the VIM text editor?</i>	138	0.180
* = significant at $p < 0.01$ after Bonferroni correction		

Table 1. Average difference between the KA output and top websites for the eleven questions (positive indicates higher ratings for KA, negative indicates higher ratings for the competing website). Each rating was an aggregate of 6 questions on a 7-point Likert scale.

the KA approach, and suggests that adding support for more structured answers (e.g., including sequential steps) could be valuable future work.

As an additional external evaluation, for the two questions (Q6 and Q7) related to automotive systems we compared the discovered categories from the KA system with two commercial knowledge service products generated by expert technicians. We compared the KA response’s accuracy and comprehensiveness, and found that it discovered all the categories referred to in these two commercial products for each question. Furthermore, the categories from the KA output provided more categories not mentioned in the commercial product (average 2.5 categories from two commercial products, while average 9.5 categories from KA). We validated these additional categories with expert automotive professionals who evaluated them as also being plausible and reasonable for the given questions. There was one instance in which two distinct categories (Encoder Motor and Encoder Motor Sensor) from the commercial products were clustered into the single category named Encoder Motor Assembly in the KA output. However, the full text answer from the KA system for Encoder Motor Assembly did still contain these two sub-components with different repair procedures.

It may seem surprising that KA would work well for questions such as automotive error codes, where the response relies heavily on technical knowledge and jargon. On further inspection we believe this is because there are many online resources that have valuable information pertaining to these questions but are in unstructured and dialog oriented forms. Workers in the sourcing phase found rich sources of online

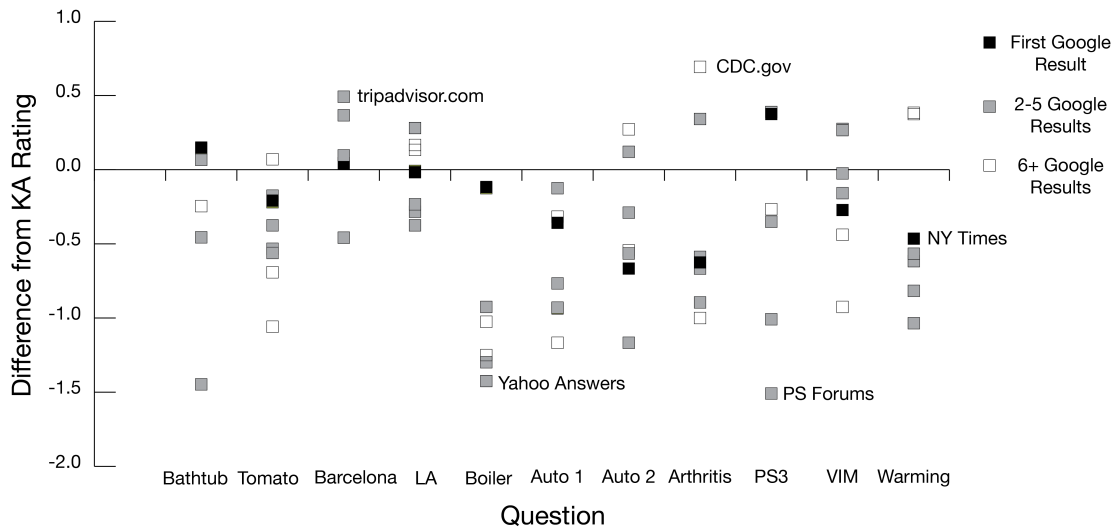


Figure 6. Results across questions and websites. Points represent the average aggregate score difference between the KA answer and an existing site

information from many car enthusiast discussion forums, in which members tried to diagnose and help each other solve their automotive problems. Although crowd workers may not understand the esoteric jargon of the automotive domain, their understanding of grammar, semantics, and argument structure was sufficient to let them find, filter, cluster, integrate, and edit this domain-specific information. These results suggest an interesting avenue for future research leveraging human understanding of semantics and argument structure to extend crowdsourcing to process expert domain knowledge and to understand the limits of where such an approach breaks down.

On average, running a question through the KA system cost a total of \$108.50 (see Table 2). Although our primary goal was to establish a proof of concept of accomplish big picture thinking in small pieces, we return to the issue of cost in the Discussion. From the self-report crowdworker feedback, workers mostly found the tasks to be easy to complete, with the clustering phase having the most difficult task.

Phase	Task Pay	Avg. # of Tasks	Avg. Cost
Sourcing	\$0.25	15	\$3.75
Clipping	\$0.50	21.6	\$10.80
Clustering 1	\$1.00	10	\$10.00
Clustering 2	\$1.00	10	\$10.00
Integrate	\$0.50	37.2	\$18.60
Edit 1	\$0.75	28.8	\$21.60
Edit 2	\$1.00	28.8	\$28.80
Images	\$0.50	9	\$4.50
Total		160.4	\$108.05

Table 2. Average number of worker tasks and average cost per phase, and overall, to run a question.

DISCUSSION

Our primary goal was to investigate the opportunities and limitations of accomplishing big thinking in small pieces, using a distributed information synthesis task as a probe. We instantiated our design approach in a prototype system called the Knowledge Accelerator which crowdsourced the process

under the constraint that no single task would pay more than \$1, and investigated its performance across a variety of complex information seeking questions. Results suggested that the output of the system compared favorably to top information sources on the web, approaching or exceeding perceived quality ratings for even highly curated and reputable sources.

The strong performance of the system is perhaps surprising given that its output was generated by many non-expert crowd workers, none of whom saw the big picture of the whole. We do not believe that this should be interpreted as a replacement for expert creation and curation of content. Instead, the power of the system may actually be attributable to the value created by those experts by generating content which the crowd workers could synthesize and structure into a coherent digest. This explanation suggests that the approach would be most valuable where experts generate a lot of valuable information that is unstructured and redundant, such as the automotive questions in which advice from car enthusiasts was spread across many unstructured discussion forums. In contrast, KA’s output did not outperform top web sources for topics such as travel, where there are heavy incentives for experts to generate well structured content. We believe its performance is likely due to its aggregation of multiple expert viewpoints rather than particularly excellent writing or structure per se, though this is a fruitful area for future investigation.

In developing the KA system, we explored a number of approaches that did not work. We initially tried to avoid a clustering phase altogether by exploring variations of the clipping task in which we provided additional context to workers in having them read through multiple sources, engage the workers who found sources in doing the clipping, or have them build on the categories that other workers had already generated rather than work independently. However, in all cases workers did not generate good labels due to a lack of context. We then explored introducing an additional “conductor” view, in which workers could be recruited as clips came in to

organize those clips and close categories that had a sufficient number of clips; however, this also failed because the conductors did not have sufficient global context to create good categories. These failures motivated the hybrid crowd-machine clustering phase.

Development of the integration and editing phases also included many false starts due to the opposite problem of giving workers *too much* context. Our first integration interface enabled multiple workers at the same time to easily view and expand all the clips in a category for within-category context, and also see the current state of how other categories were developing for between-category context. Our idea was that as workers integrated clips and built out more options exposure to the other clips and options in real time would help them create more coherent digests. However, this approach proved overwhelming for scaling up to a large number of crowd workers engaged for short time periods. This motivated us to split up within-category and across-category consistency into the integration and editing phases and the development of the vote-edit pattern.

We encountered a number of places where our approach could be improved. As evidenced in the VIM question, the lack of support for nuanced structure in our digests can prove problematic. For some sources such as tutorials or how-tos, supporting sequential dependencies between steps could be useful. While our output was able to support such dependencies in an ad-hoc way within a category (such as the sequential steps for plunging a drain) it would be profitable to be able to support sequential dependencies across categories (e.g., first try x, then try y). More structure could also be beneficial for particular domain areas, such as explicitly capturing symptoms and causes as different types for automotive or medical diagnostic questions.

The system could also benefit from including iteration. For example, after workers completed the integration phase they were asked the question “What else needs to be done to make this a complete answer?”. While many obviously said the section needed be edited, one of the most popular responses was “Needs more information.” This suggested to us that while our clips and categories had pulled in most of the information, there was more information in some sections we were missing. One possibility is to introduce an iterative component at this point – as workers are integrating information into the pad and notice missing information, they can request for other workers to go out and find that additional information through clipping. Thus while the system was partially successful at taking a breadth-oriented approach rather than the deeply iterative approach typical of sensemaking [20, 22, 58, 61], understanding how to best incorporate iteration would be a valuable area for future work.

A final area for future improvement is the cost associated with producing answers. Our digests took approximately \$100 to produce. While intended as a proof-of-concept prototype and similar in scale to other such crowdsourcing systems [3, 17], it is interesting to consider what could be done to move the approach towards a useful production system with lowered costs. One area of improvement is optimization: by dynam-

cally deciding how many workers and products to use in each stage final costs could be dropped significantly (e.g., as in [30]). Furthermore, for many practical information seeking purposes the categories and associated clips may be sufficient, which would obviate the need for the expensive stages of integration and editing and reduce costs by over 65%.

Perhaps the most interesting possibility is if answers could be reused across questions. Although users have complex information seeking needs, many of the queries they issue are similar. For example, a recent study estimated that 3% of search queries account for 13 of total search volume [66]. Thus at a minimum, many answers could be amortized across users with the same question. A particularly promising but challenging opportunity is if similar questions may be able to reuse components of already summarized answers; for example, a question on investing advice for a 50 year old might use some common categories as for a 20 year old, but others would be unique to the new question’s context. Challenges for the reuse of information are how the system would be able to identify the similarity for possible answers during each information synthesis phase and what level of granularity should be considered to for an effective system. Spatial and temporal reasoning over the existing knowledge and new information could be considered to provide context-aware and up-to-date answers.

We hope the design choices embodied in the KA prototype system and the design patterns discussed here may be useful for other system designers working to distribute cognitive complex tasks. Some domains that might benefit from this include microtask markets, which could benefit from supporting more complex tasks; volunteer crowdsourcing efforts such as Wikipedia [35] or friendsourcing in which many small contributions are readily available [10]; or self-sourcing in which the crowd within could accomplish complex tasks in small increments (e.g., waiting for the bus) without needing to load the entire task context into working memory [63]. Overall, we believe this approach represents a step towards a future of big thinking in small packages, in which complex and interdependent cognitive processes can be scaled beyond individual cognitive limitations by distributing them across many individuals.

ACKNOWLEDGMENTS

The authors would like to thank to Andrew Peters for his initial work on the KA system. This work was supported by NSF grants IIS-1149797, IIS-0968484, IIS-1111124, Bosch, and Google.

APPENDIX

See <http://nhahn.org/portfolio/ka.html#appendix> for additional resources, including the original articles generated by the KA system.

REFERENCES

1. 2015. Etherpad Lite. <https://github.com/ether/etherpad-lite>. (2015).
2. Salman Ahmad, Alexis Battle, Zahan Malkani, and Sepander Kamvar. 2011. The jabberwocky programming environment for structured social computing. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 53–64.
3. Paul André, Aniket Kittur, and Steven P Dow. 2014a. Crowd synthesis: Extracting categories and clusters from complex data. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 989–998.
4. Paul André, Robert E Kraut, and Aniket Kittur. 2014b. Effects of simultaneous and sequential work structures on distributed collaborative interdependent tasks. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 139–148.
5. Anonymised. 2016. Alloy: Clustering with Crowds and Computation. In *submission to CHI* (2016).
6. Michelle Q Wang Baldonado and Terry Winograd. 1997. SenseMaker: an information-exploration interface supporting the contextual evolution of a user’s interests. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 11–18.
7. Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 550–557.
8. Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. 2011. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 33–42. DOI : <http://dx.doi.org/10.1145/2047196.2047201>
9. Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010a. Soylent: a word processor with a crowd inside. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. ACM, 313–322.
10. Michael S Bernstein, Desney Tan, Greg Smith, Mary Czerwinski, and Eric Horvitz. 2010b. Personalization via friendsourcing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 2 (2010), 6.
11. Michael S Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, and Eric Horvitz. 2012. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 237–246.
12. Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and others. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. ACM, 333–342.
13. Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 335–336.
14. Pew Research Center. 2015. Generational differences in online activities. Report. (25 July 2015). <http://www.pewinternet.org/2009/01/28/generational-differences-in-online-activities/>.
15. Dana Chandler and Adam Kapelner. 2013. Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization* 90 (2013), 123–133.
16. Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 27.
17. Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. 2013. Cascade: Crowdsourcing Taxonomy Creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1999–2008. DOI : <http://dx.doi.org/10.1145/2470654.2466265>
18. Richard L Daft and Karl E Weick. 1984. Toward a model of organizations as interpretation systems. *Academy of management review* 9, 2 (1984), 284–295.
19. Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. 2007. Overview of the TREC 2007 Question Answering Track.. In *TREC*, Vol. 7. 63.
20. Brenda Dervin. 1983. *An overview of sense-making research: Concepts, methods, and results to date*. The Author.
21. Brenda Dervin. 1992. From the minds eye of the user: The sense-making qualitative-quantitative methodology. *Qualitative research in information management* 9 (1992), 61–84.
22. Brenda Dervin. 1998. Sense-making theory and practice: an overview of user interests in knowledge seeking and use. *Journal of knowledge management* 2, 2 (1998), 36–46.
23. Günes Erkan and Dragomir R Radev. 2004. LexRank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* (2004), 457–479.
24. Dennis A Gioia and Kumar Chittipeddi. 1991. Sensemaking and sensegiving in strategic change initiation. *Strategic management journal* 12, 6 (1991), 433–448.
25. Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization-Volume 4*. Association for Computational Linguistics, 40–48.

26. Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence* 2, 3 (2010), 258–268.
27. Udo Hahn and Ulrich Reimer. 1999. Knowledge-based text summarization: Saliency and generalization operators for knowledge base abstraction. *Advances in Automatic Text Summarization* (1999), 215–232.
28. Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.
29. Jim Jones. 2013. Turkee Ruby Gem. <https://github.com/aantix/turkee>. (2013).
30. Ece Kamar, Severin Hacker, and Eric Horvitz. 2012. Combining Human and Machine Intelligence in Large-scale Crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS '12)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 467–474. <http://dl.acm.org/citation.cfm?id=2343576.2343643>
31. Ece Kamar and Eric Horvitz. 2013. Light at the End of the Tunnel: A Monte Carlo Approach to Computing Value of Information. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 571–578. <http://dl.acm.org/citation.cfm?id=2484920.2485011>
32. Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 4017–4026.
33. Aniket Kittur, Ed H Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 453–456.
34. Aniket Kittur, Susheel Khamkar, Paul André, and Robert Kraut. 2012. CrowdWeaver: Visually Managing Complex Crowd Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1033–1036. DOI : <http://dx.doi.org/10.1145/2145204.2145357>
35. Aniket Kittur and Robert E Kraut. 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 37–46.
36. Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013a. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1301–1318.
37. Aniket Kittur, Andrew M Peters, Abdigani Diriye, Trupti Telang, and Michael R Bove. 2013b. Costs and benefits of structured information foraging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2989–2998.
38. Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 43–52.
39. Aniket Kittur, Bongwon Suh, Bryan A Pendleton, and Ed H Chi. 2007. He says, she says: conflict and coordination in Wikipedia. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 453–462.
40. Gary Klein, Brian Moon, and Robert R Hoffman. 2006. Making sense of sensemaking 2: A macrocognitive model. *Intelligent Systems, IEEE* 21, 5 (2006), 88–92.
41. Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 1003–1012.
42. Anand P Kulkarni, Matthew Can, and Bjoern Hartmann. 2011. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2053–2058.
43. Walter Lasecki, Christopher Miller, Adam Sadilek, Andrew Abumoussa, Donato Borrello, Raja Kushalnagar, and Jeffrey Bigham. 2012. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 23–34.
44. Walter S Lasecki, Christopher D Miller, and Jeffrey P Bigham. 2013a. Warping time for more effective real-time crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2033–2036.
45. Walter S Lasecki, Christopher D Miller, Raja Kushalnagar, and Jeffrey P Bigham. 2013b. Legion scribe: real-time captioning by the non-experts. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM, 22.
46. Edith Law and Haoqi Zhang. 2011. Towards Large-Scale Collaborative Planning: Answering High-Level Search Queries Using Human Computation.. In *AAAI*.
47. Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. 2010. Turkkit: human computation algorithms on mechanical turk. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. ACM, 57–66.
48. Kurt Luther, Casey Fiesler, and Amy Bruckman. 2013. Redistributing Leadership in Online Creative Collaboration. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1007–1022. DOI : <http://dx.doi.org/10.1145/2441776.2441891>
49. Inderjeet Mani and Eric Bloedorn. 1997. Multi-document summarization by graph search and matching. *arXiv preprint cmp-lg/9712004* (1997).

50. Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Commun. ACM* 49, 4 (April 2006), 41–46. DOI : <http://dx.doi.org/10.1145/1121949.1121979>
51. Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI*. 453–460.
52. James McKinney. 2015. TfIdfSimilarity Ruby Gem. <https://github.com/jpmckinney/tf-idf-similarity>. (2015).
53. Frances J Milliken. 1990. Perceiving and interpreting environmental change: An examination of college administrators' interpretation of changing demographics. *Academy of management Journal* 33, 1 (1990), 42–63.
54. Meredith Ringel Morris, Jarrod Lombardo, and Daniel Wigdor. 2010. WeSearch: supporting collaborative search and sensemaking on a tabletop display. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 401–410.
55. Aditya Parameswaran, Ming Han Teh, Hector Garcia-Molina, and Jennifer Widom. 2013. Datasift: An expressive and accurate crowd-powered search toolkit. In *First AAAI Conference on Human Computation and Crowdsourcing*.
56. Sharoda A Paul and Meredith Ringel Morris. 2009. CoSense: enhancing sensemaking for collaborative web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1771–1780.
57. Sharoda A Paul and Madhu C Reddy. 2010. Understanding together: sensemaking in collaborative information seeking. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 321–330.
58. Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.
59. Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S Bernstein. 2014. Expert crowdsourcing with flash teams. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 75–85.
60. Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. 2011. An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets.. In *ICWSM*.
61. Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. The Cost Structure of Sensemaking. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 269–276. DOI : <http://dx.doi.org/10.1145/169059.169209>
62. Bongwon Suh, Gregorio Convertino, Ed H Chi, and Peter Pirolli. 2009. The singularity is not near: slowing growth of Wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. ACM, 8.
63. Jaime Teevan, Daniel J Liebling, and Walter S Lasecki. 2014. Selfsourcing personal tasks. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2527–2532.
64. Karl E Weick. 1964. Reduction of cognitive dissonance through task enhancement and effort expenditure. *The Journal of Abnormal and Social Psychology* 68, 5 (1964), 533.
65. Karl E. Weick. 1995. *Sensemaking in organizations*. Vol. 3. Sage.
66. Ryen W White, Mikhail Bilenko, and Silviu Cucerzan. 2007. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 159–166.
67. Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. 2012. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 217–226.