Identifying Redundancy and Exposing Provenance in Crowdsourced Data Analysis

Wesley Willett, Shiry Ginosar, Avital Steinitz, Björn Hartmann, and Maneesh Agrawala

Abstract— We present a system that lets analysts use paid crowd workers to explore data sets and helps analysts interactively examine and build upon workers' insights. We take advantage of the fact that, for many types of data, independent crowd workers can readily perform basic analysis tasks like examining views and generating explanations for trends and patterns. However, workers operating in parallel can often generate redundant explanations. Moreover, because workers have different competencies and domain knowledge, some responses are likely to be more plausible than others. To efficiently utilize the crowd's work, analysts must be able to quickly identify and consolidate redundant responses and determine which explanations are the most plausible. In this paper, we demonstrate several crowd-assisted techniques to help analysts make better use of crowdsourced explanations: (1) We explore crowd-assisted strategies that utilize multiple workers to detect redundant explanations. We introduce *color clustering with representative selection*—a strategy in which multiple workers cluster explanations and we automatically select the most-representative result—and show that it generates clusterings that are as good as those produced by experts. (2) We capture explanation provenance by introducing highlighting tasks and capturing workers' browsing behavior via an embedded web browser, and refine that provenance information via source-review tasks. We expose this information in an explanation-management interface that allows analysts to interactively filter and sort responses, select the most plausible explanations, and decide which to explore further.

Index Terms—Crowdsourcing; Social Data Analysis

1 Introduction

As analysts, data-journalists, and other data consumers explore and examine large datasets, they must often consider many different slices of their data and engage in an iterative process of sensemaking, involving both visualizations and outside resources [10]. Generating explanations for trends and outliers in datasets and finding evidence to support those explanations are key parts of this sensemaking loop and can entail considerable work for the analyst (examining charts, ideating, finding and confirming sources, etc.). Given a large dataset—for example, a journalist examining new employment figures from hundreds of cities or inspecting data about thousands of contributors to a political campaign—it may not be feasible for a single analyst or even a small team to examine all of the relevant views and generate potential explanations for outliers or trends.

However, analysts can expedite this process by asking paid crowd workers—drawn either from an external crowd marketplace like Amazon Mechanical Turk (`www.mturk.com`) or from an internal pool of trusted collaborators—to perform large numbers of small exploratory analysis tasks. Working in parallel, crowd workers can help identify important views, generate diverse sets of explanations for trends and outliers, and—in some cases—even provide important domain expertise that the analyst lacks. This approach complements the existing individual and team-based models of analysis currently used by data analysts and may provide a valuable tool for expediting the visual analysis of large data sets across a wide range of different disciplines.

Prior work by Willett et al. [19] has demonstrated that crowd workers can produce good explanations and resources for a range of public-interest datasets. However, that work also identifies a weakness of the approach—namely that eliciting large volumes of explanations and observations from the crowd may *create* additional work for analysts who must examine and integrate them. For example, large numbers of workers operating in parallel often produce many redundant responses that give the same general explanation for a trend or outlier. Analysts must spend time filtering and condensing these responses to identify unique explanations and determine if redundant explanations corroborate one another. Additionally, because individual workers have different competencies and domain knowledge, some of the explanations they produce are more plausible—more likely to be true—than others. Determining which explanations are plausible and which are not can also be difficult, in part, because explanations generated by workers often lack detailed provenance—information about the sources used to produce the explanation. In these cases, analysts cannot determine whether a worker's explanation is derived from a reputable source or is the worker's own speculation.

This paper focuses on integrating crowd-based contributions into analysts' workflows by providing tools that help analysts process, filter, and make sense of the explanations and insights generated by crowd workers. First, we identify a set of criteria (*text clarity and specificity*, *explanation redundancy*, and *explanation provenance*) that analysts can use to filter and organize sets of explanations and decide whether or not they are plausible. We then contribute two sets of techniques designed to help analysts assess the redundancy and provenance of crowdsourced explanations:

(1) We explore strategies for identifying and grouping redundant explanations and introduce *color clustering with representative selection*, a novel crowd-assisted clustering method. Using this approach—in which multiple workers cluster explanations manually and an algorithm selects the most-representative clustering—our system can group small sets of explanations into clusters that are similar to those produced by experts.

(2) We help analysts gauge the plausibility of explanations by exposing more detailed explanation provenance. We introduce highlighting tasks that allow workers to make finer-grained citations and capture workers' browsing behavior via an embedded web browser. We also show how workers can help verify the provenance of others' explanations via source-checking tasks.

Finally, we demonstrate an explanation-management interface that allows analysts to interactively explore clustered explanations and examine their provenance. Using this interface, analysts can quickly group and filter responses, in order to determine which explanations should be further considered.

- *Wesley Willett is with INRIA. E-mail: wesley.willett@inria.fr.*
- *Shiry Ginosar, Avital Steinitz, Björn Hartmann, and Maneesh Agrawala are with UC Berkeley E-mail: shiry, steinitz, bjoern, magrawala@eecs.berkeley.edu.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2 Related Work

Over the past decade, a number of social data analysis systems, including Sense.us [7], Many Eyes [17], and Pathfinder [8] have allowed large numbers of users to share, explore, and discuss data visualizations. These research systems, (as well as commercial social data analysis systems like Data360.org and the now-defunct Swivel.com) were premised on the idea that many users operating together can parallelize the process of visual data analysis. While motivated users of these tools do explore and discuss datasets extensively [18], most datasets receive little attention and it is difficult for analysts to leverage other users' efforts in a consistent and systematic way.

However, evaluations of systems like CommentSpace [20] have shown that dividing visual analysis tasks into smaller, explicit stages where participants generate explanations, then organize and build on them, can produce good analytic results, at least in controlled settings. Willett, et al. [19] break visual analysis into even smaller explanation-generation microtasks and allows analysts to systematically assign them to crowd workers. Analysts can incorporate these tasks into existing workflows, providing an easy way to generate many possible explanations for trends, outliers, and other features of datasets. This approach is similar to other human computation systems like Soylent [3], VizWiz [4], and Mobi [23] that embed crowd labor into existing tools and workflows. We explore additional techniques to help analysts leverage crowds during the next phase of their analysis, when they need to evaluate large numbers of candidate explanations.

Other recent work has also explored the application of distributed collaborators to sensemaking and analysis tasks. Fisher, et al. [6] examine how distributed collaborators iteratively build an understanding of information by organizing their work as shared knowledge maps. While they focus on small groups of collaborators performing open-ended tasks that are less data-oriented, we apply larger groups of paid workers to perform small, well-defined pieces of analysis tasks.

At a more fundamental level, Yi et al.'s CrowdClustering [22], Tamuz et al.'s "crowd kernel" [14], and Chilton et al.'s Cascade [5], have recently explored the use of paid crowds to cluster images and build text taxonomies. We explore an alternate crowd-based clustering method that produces good results for sets containing tens of explanations and describe how these other existing clustering techniques complement our contributions.

Researchers have also explored "instrumenting the crowd" [11] by augmenting crowd-based tasks to track workers' behavior and automatically assess the quality of their work. We also log worker activity, but use it to help analysts assess explanation provenance.

3 Crowdsourcing Data Analysis

We utilize a data analysis workflow (Figure 1) that extends the one proposed by Willett et al. [19]. As in the original workflow, analysts use crowd workers to help generate candidate explanations for trends in a new dataset then rate, cluster, and verify them. Consider the example of a data-journalist who has just gained access to detailed employment statistics for several hundred US cities. This analyst may wish to quickly find cities where unemployment trends differ from the national average and explore possible explanations for those differences. Using this workflow, the analyst either manually or automatically (using a statistics package) selects views of the dataset that contain outliers, trends, or other features of interest. The analyst then submits these charts as *analysis microtasks* to workers in a paid crowd marketplace. Each microtask (Figure 2) displays a single chart along with a series of prompts. Workers examine each chart, generate candidate explanations for why the trend or outlier in it may have occurred, and provide links to web pages that support their explanations.

Willett et al. have demonstrated that, using a simpler version of this workflow, it is possible to produce high-quality explanations for a range of datasets. However, once workers produce explanations, it is still up to an analyst to examine each one to determine if it is plausible and if she should explore it further. Because workers often generate many candidate explanations for each chart, identifying the most promising ones is time-consuming and entails considerable effort from the analyst.

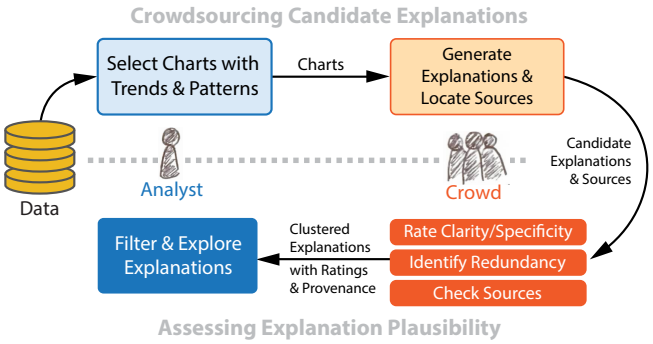

Fig. 1. Our workflow for crowdsourcing social data analysis (modified from [19]). First, an analyst generates a large number of possible charts from a data set (top-left). Crowd workers then provide candidate explanations for trends and patterns (top-right). Next, workers help analysts assess explanation plausibility by rating explanations, identifying redundancy, and checking sources (bottom-right). Analysts can then interactively explore the results (bottom-left). This paper focuses on the latter half of the workflow.

We extend this workflow to help analysts more easily determine which explanations are the most plausible. Specifically, we introduce crowdsourced techniques to help analysts identify and prioritize sets of redundant answers and quickly assess the plausibility of the possible explanations for the same phenomenon. We also demonstrate a prototype interface that allows analysts to use this information to quickly filter, scan, and annotate explanations.

4 Assessing Explanation Plausibility

When considering explanations for trends or outliers, an analyst's key task is to determine if each candidate explanation is likely to be true and decide whether it should be discarded, retained, or explored further. Based on our own experience examining several thousand crowdsourced explanations, we propose several criteria that analysts can use to help assess the plausibility of explanations. These criteria (enumerated below as C_1 through C_3) motivate the design of our clustering and provenance tools, as well as the specific features of our explanation management interface.

C_1: Text Clarity and Specificity. Some fraction of workers in crowd marketplaces typically satisfice—performing the bare minimum amount of work to complete the task—and may generate poorly-constructed, unspecific, or logically implausible results. By comparison, we observe that clear and specific explanations that appear internally consistent are often more likely to be correct.

C_2: Explanation Redundancy. We observe that, if an explanation is proposed multiple times by different workers, it may indicate that the explanation is widely accepted and merits further investigation. Conversely, a lack of redundant explanations may signal that there are many possible answers, and the odds that the workers have found the most plausible one are lower [16]. Clustering redundant explanations and indicating the frequency with which each explanation occurs can help analysts make these assessments.

C_3: Explanation Provenance. An analyst can also use information about the source from which an explanation was taken, in order to help determine if it is plausible. To make this judgment, we find that it helps to understand both where the explanation originated and how it was collected or generated by the worker. Understanding an explanation entails several kinds of considerations:

$C_{3.1}$: Does the explanation cite a reputable source? If an explanation draws from a source the analyst is familiar with, the analyst can also leverage his or her knowledge of the source to help decide if an explanation is plausible. A citation from a recognized and trusted source (for example a known news organization or reference) usually bolsters

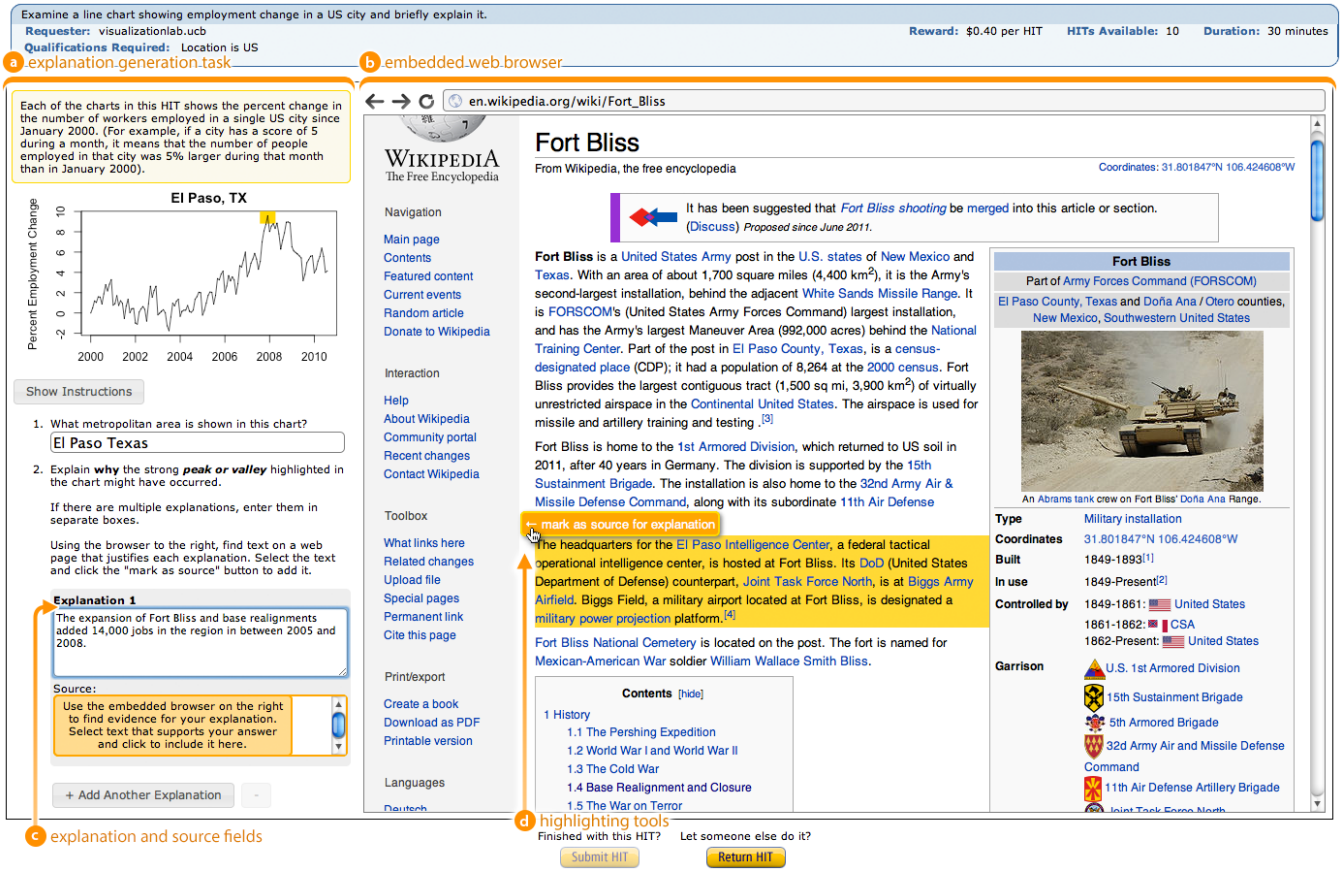

Fig. 2. An analysis microtask (A) is paired with an proxied web browser embedded inside the HIT (B). The explanation prompts in the interface (C) are linked to highlighting tools (D) that let workers cite specific sections of source documents.

an explanation's plausibility, while an unknown or disreputable source may diminish it. Surfacing details about the cited source and other resources used by the worker as they derived the explanation can help analysts make this assessment.

$C_{3.2}$: *Does the content of the explanation come from the source or the worker?* In our experience, workers who are not domain experts (including most workers on crowd marketplaces like Mechanical Turk) are more adept at pulling good explanations from sources than producing them independently. As a result, explanations that repeat or paraphrase facts and inferences from a trusted source tend to be more credible than explanations based on facts or inferences produced entirely by the worker. As such, an indication of whether or not the content is copied or paraphrased directly from the source can help analysts assess plausibility more easily.

$C_{3.3}$: *Is the explanation corroborated by multiple sources?* If multiple versions of the explanation cite the same source, it indicates a reliance on that source. If the source is known and trusted, this reliance can increase confidence in the explanation. Alternatively, if multiple explanations cite an unknown source, it can suggest that the source is one that the analyst may wish to consider directly. Finally, multiple versions of an explanation that cite *different* reputable sources may increase confidence even further, since these sources can corroborate one another [21].

4.1 Assessing Text Clarity and Specificity

Our workflow helps analysts assess the clarity and specificity of explanations by showing them to a second set of workers as *rating microtasks*. Workers assign a 0-5 score that indicates how clear and specific each response is and whether it answers the prompt. Willett et al. have shown that this approach can separate clear and specific explanations from unclear ones [19].

4.2 Identifying Redundancy via Crowdsourcing

Grouping frequently-recurring explanations together can keep analysts from spending time considering duplicate explanations and can help analysts see which explanations are the most frequent or are corroborated by multiple sources. However, determining whether multiple explanations are redundant is a difficult and somewhat subjective task.

A wide range of automated text similarity and topical clustering methods are available for grouping and labeling pieces of text [9]. However, automated approaches tend to rely on the assumption that similar explanations use similar language. These measures of explanation similarity can fail when explanations use different terms to describe the same phenomenon (e.g., "layoffs" instead of "downsizing") or when the connection between two comments requires outside knowledge (e.g., the notion that widespread "layoffs" may be related to an "economic downturn"). Moreover workers' explanations are typically short and the total number of explanations for a single feature can be small. Short explanations present a challenge for text similarity algorithms, since their overlapping content tends to be very sparse, making it difficult to produce reliable clusters [12].

In contrast to automated approaches, human workers can leverage semantic information and outside knowledge to cluster sets of textual explanations. However, individual workers can only examine a limited number of explanations at one time. Workers may also cluster explanations differently from one another, making it challenging to integrate clusterings from multiple workers. As a result, crowd-based clustering approaches can benefit from distributing the clustering tasks across workers or otherwise combining their effort.

One common approach to crowd-based clustering is what we call *distributed comparision*, in which multiple workers compare pairs of responses and indicate whether they give the same explanation. The system then aggregates these similarity judgements and uses them to cluster the complete set of explanations. Comparing pairs of responses is a straightforward task that can be easily distributed across workers.

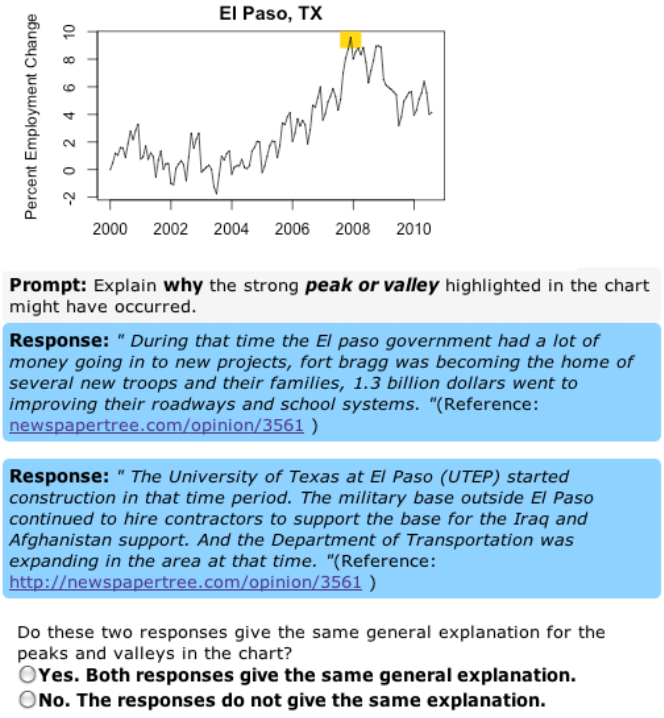

Fig. 3. In our *distributed comparison* implementation, we show workers pairs of explanations for a phenomenon and ask them to decide whether or not the two explanations are redundant.

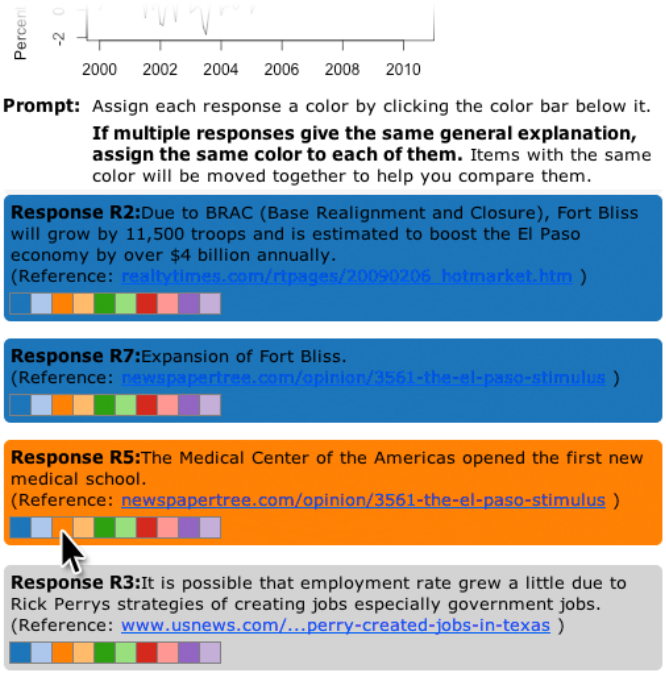

Fig. 4. In the manual *color clustering* approach, we show workers all explanations for a trend or outlier and ask them to create clusters by marking redundant explanations with the same color. Similarly-colored explanations are grouped together on-screen, allowing workers to see their clusters in context.

However, clustering using workers' judgements can be difficult, in part because it is difficult to assess how much variance is acceptable within a cluster and how many clusters a given set should be grouped into.

We introduce *color clustering with representative selection*, a crowdsourced approach for clustering explanations in which multiple workers consider all of the responses and use a color-coding interface to interactively organize them into clusters. Our system automatically selects the best clustering from amongst those produced by the workers. This selection algorithm allows us to obtain a single, consistent clustering authored by one worker but validated by the work of others.

4.2.1 Distributed Comparison

Existing crowd-based clustering methods, including Yi et al.'s Crowd-Clustering [22] and Tamuz et al.'s "crowd kernel" [14] break clustering tasks into small, two- or three-way comparison tasks which are used to construct a similarity matrix describing the resemblance between items(typically images) in a set. The resulting similarity matrix can be used to cluster the items into discrete categories using techniques like k-means clustering. We refer to these approaches, in which multiple users compare items a few at a time and their results are aggregated to produce clusterings as *distributed comparison*.

As a baseline for comparison, we implemented a variant of this approach (Figure 3). In our *distributed comparison* implementation, we ask crowd workers to examine pairs of explanations and indicate whether or not they are redundant. Using multiple workers, we collect at least 5 judgments for every pair of explanations, then average the binary similarity judgments to produce an average similarity score for the pair. To limit the impact of workers who attempt to game the task, we include pairs of gold standard explanations with known similarity, and remove results from workers who fail to mark them correctly. We then use the remaining similarity scores to group the explanations in to a fixed number of clusters using k-means clustering.

While distributed comparison decomposes clustering into small tasks that are easy for workers to perform, it scales poorly as the number of explanations increases. Assessing redundancy for all n pairs requires $\binom{n}{2}$ operations and grows quadratically as the number of explanations increases. To reduce the total number of comparisons, Tamuz et al. frame these tasks as triplet-based comparisons and sample to build a partial similarity matrix. [14]. Meanwhile, Yi et al. [22] use matrix completion to build similarity matrices without asking workers to compare all pairs of items. However, both of these approaches create approximations of the complete worker-generated similarity matrix, and may produce similarity scores for some pairs that were not intended by workers. As a result, we opted to build the complete similarity matrix by eliciting multiple worker comparisons for every pair of explanations.

One challenge when using k-means or other similar methods is picking the number of target clusters, k. Tibshirani et al. [15] provide an overview of a number of heuristics for selecting k. However, because the proportion of redundant explanations can vary from set to set and is dependent on the semantics of the data, a general method for selecting k for all data sets is unlikely to exist in practice. In our experience all of the methods suggested by Tibshirani et al. suggested cluster sizes that deviated widely from our own manual clusterings. Although choosing good cluster sizes for workers results remains an unsolved problem, our implementation allows us to evaluate the best-case performance of distributed comparison by clustering using all possible values of k and selecting the best.

Pairwise comparisons are also problematic because workers never see all of the explanations at once and may miss redundancies that require context from other explanations in the set. For example, four responses attributing employment growth in El Paso to (A) *"a new medical complex"*, (B)*"a new medical center at UTEP"*, (C) *"construction on the university campus"*, and (D) *"constructions of new building on campus"* might be split into two separate clusters if considered in isolation. If presented as a series of binary comparisons, workers are likely to group A and B together because they both mention the medical complex, and are likely to group C and D because they discuss university construction. However, seeing the larger set of explanations together could give a worker the opportunity to realize that all four explanations are actually attributing growth to the same hospital construction project.

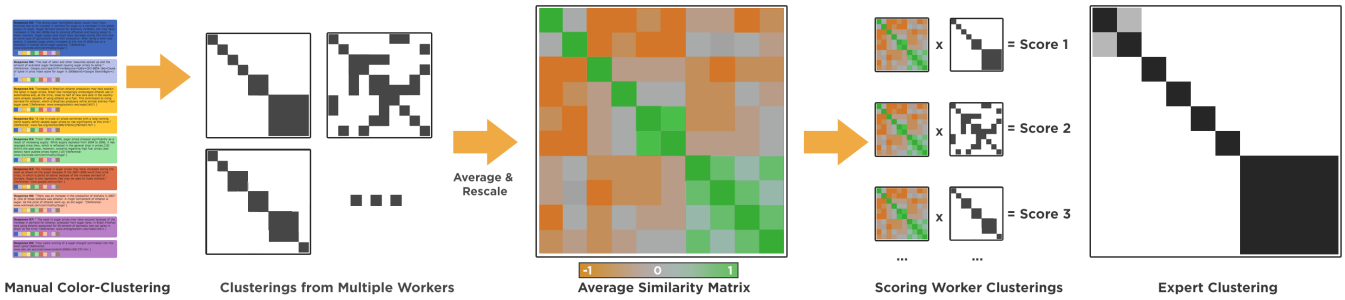

Fig. 5. Illustration of our algorithm to select good worker clusterings from a larger set of possible ones. The algorithm transforms workers' color clusterings (left) into similarity matrices (center-left) and aggregate them to create an average similarity matrix (center) — which is normalized here to improve readability. It then compares individual clusterings against the average to choose the most-representative (center-right), which typically resembles clusterings generated by experts (right).

4.2.2 Color Clustering with Representative Selection

Due to the issues associated with distributed comparison, we developed an alternate clustering approach in which workers examine all of the explanations for a chart and group them manually. Displaying the full set of explanations gives workers the opportunity to identify clusters (like the one described above) that may not be obvious without additional context.

To simplify the task of specifying clusters, we created a system where workers group comments by color-coding them. In each manual color clustering task (Figure 4), workers see the full set of explanations for a chart and can color-code each explanation by clicking in the palette attached to it. When a worker assigns the same color to multiple responses, the system moves the responses next to one another, creating visually distinct clusters. These clusters allow workers to see their clusters as they create them and compare similar comments side by side without having to rely as strongly on their working memory.

Clustering explanations is a subjective task and the boundaries between clusters can vary depending on subtle interpretations of the explanation text. As a result, multiple workers—even well-intentioned and well-informed ones—may produce different clusterings. Because many different clusterings may be valid, it is difficult to identify one clustering as the most correct or to combine the clusterings produced by multiple workers into a single clustering.

To design an algorithm for selecting the best clustering from a set, we built on several observations:

(1) If most workers agree that a particular group of explanations should be clustered together, there is a high likelihood that that grouping indeed reflects similarities in the explanations' content [13]. As a result, we assume that the clusterings that are the most dissimilar from all other clusterings for a given chart are more likely to be badly clustered, while the clusterings that are the most similar to all the others are likely to be clustered well.

(2) Most systematic errors (e.g., a worker satisficing by lumping all explanations into a single cluster) can be caught by including gold-standard tests and by eliminating workers who complete the task in less time than it would take for a fast reader to parse all of the explanations. Other errors tend to be noisy (e.g., a worker satisficing by randomly clusterings explanations) and are not usually duplicated by multiple workers.

(3) A single worker's clustering is more likely to be internally consistent and understandable to the analyst, because it reflects a single set of judgments made in-context with one another. Therefore, choosing a single worker's clustering is preferable to combining results from multiple workers.

Based on these observations, we designed a procedure (which we call *color clustering with representative selection*) for extracting the *most-representative* clusterings from a set of manual color clusterings generated by multiple workers (Figure 5). The rating scheme we use is based purely on the correspondences between workers' clusterings, rather than on the content or quality of the explanations.

First, we ask multiple workers to cluster the explanations for a chart using the manual color clustering interface (Figure 5 left). We then construct a separate *cluster similarity matrix* for each worker's clustering (Figure 5 center-left). Each row and column in this matrix corresponds to one of the explanations in the set. We initialize all elements in this matrix to 0, then assign a 1 to each element where the worker placed the explanation on the corresponding row and the explanation on the corresponding column into the same cluster.

Next, we average together the matrices from all of the workers who clustered the set. This operation produces a single *average similarity matrix* (Figure 5 center). Larger values in this matrix correspond to pairs of explanations that were clustered together by the majority of workers, while smaller values correspond to pairs that the majority of workers did not put in the same cluster.

Finally, we select the *most-representative* clustering—the clustering from a single worker that most closely matches the average similarity matrix. We treat the values in the average similarity matrix as weights and use them to compute a weighted score for each worker's clustering. Specifically, we compute the Froebnius product of each individual worker's binary similarity matrix with the average similarity matrix by multiplying them together element-wise, then sum the values of all the elements in the product to obtain a final score for the clustering (Figure 5 center-right). We retain only the clustering which produces the highest total score. The resulting clustering is the work of a single worker, but is most strongly corroborated by the clusterings produced by the other workers.

4.3 Collecting Explanation Provenance

In addition to redundancy, analysts may also consider the reputability of the sources workers use to produce their explanations. We have developed several techniques to provide analysts with information about the sources workers use.

4.3.1 Logging Activity and Sources

We instrument the analysis microtasks that workers perform so that they provide a record of workers' browsing activity during each task. Recording the sites that workers visit as they perform microtasks is difficult to implement in practice because the same-origin policy [1] implemented by modern web browsers prevents code from one internet domain from accessing web pages loaded from other domains. As a result, our microtasks cannot monitor activity that occurs in browser windows or tabs that do not originate from our site.

We circumvent the restriction by having workers browse and search for sources using a custom web browser embedded within the analysis microtask (Figure 2B). This custom browser consists of a set of browser controls and an IFrame that loads web pages via our own custom proxy server. Requesting and then serving pages via our server (Figure 6) allows us to log each page workers visit and track any web searches they perform as they forage for sources and candidate explanations. For both technical and security reasons, we do not proxy content served using protocols other than HTTP and do not handle third-party cookies. As a result, we cannot load content from sites that require users to authenticate or log in. Additionally, we cannot guarantee that workers perform all of their browsing within our proxied

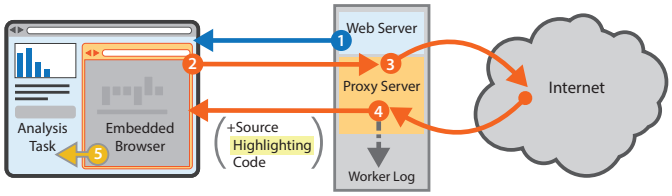

Fig. 6. In our instrumented tasks, analysis microtasks are loaded from our web server (1). When workers look for evidence using the embedded web browser inside the task, page requests are redirected via our proxy server (2). The proxy server requests pages from their source (3), then logs them and injects custom highlighting code (4). Workers can then highlight text in embedded browser to have it included directly in their explanations (5).

interface rather than in another browser window. However, our analysis of log data suggests that most of the sites workers visit are rendered appropriately via the proxy and that workers are active within our browser window for most of the time they spend on the task.

4.3.2 Supporting Fine-Grained Citations

Typically, when a worker cites a web page to support an explanation, only a small portion of the page (a paragraph or even a few sentences) is directly relevant to their explanation. Page-level citations can make it difficult for analysts or other workers to assess a source, since they may need to examine the entire web page to find the relevant text. We support finer-grained source citations by allowing workers to highlight specific blocks of text within pages as sources.

We add highlighting controls to existing web pages by injecting custom code into each page as it is delivered by our proxy server. When a worker identifies a block of text on a proxied page that provides or supports their explanation, they can highlight the text and then click on an overlay (Figure 2D) to mark it as a source. We save the selected text and the URL of the page along with the explanation.

4.3.3 Detecting Copying and Paraphrasing

Understanding whether an explanation came directly from the source or the worker can be important when assessing the plausibility of a response. In general, we know relatively little about the domain expertise of workers recruited in marketplaces like Mechanical Turk. Therefore, our default assumption is that explanations that directly paraphrase a reputable source are likely to be plausible and are more desirable for the analyst. When workers add their own ideas and inferences to an explanation, we assume the explanation is less likely to be plausible, and the analyst may wish to either disregard the explanation or check the source themselves.

While people can generally identify whether or not an explanation is derived or paraphrased from a source, paraphrasing is difficult to detect automatically. In our workflow, we use *source-checking microtasks* to determine whether or not explanations are drawn directly from a source. In these microtasks, workers examine an explanation generated by another worker, along with the source document from which they derived it, and indicate whether the explanation "is copied or paraphrased from the cited source".

5 Evaluation

To illustrate and test our strategies for clustering responses and checking source provenance, we conducted a deployment of our system using workers from Mechanical Turk. We first asked workers to generate explanations for 12 charts drawn from 3 datasets covering a range of public-interest data types (US employment data for major metropolitan areas, graduation and earning statistics for major universities, and UN food price indices). Each chart highlighted a single outlier and workers were asked specifically to explain it. We showed each chart to 10 different workers, for a total of 120 analysis microtasks. A total of 93 workers participated, producing a corpus of 156 explanations (each worker could provide more than one explanation per task).

5.1 Redundancy

To evaluate the performance of our color clustering with representative selection technique, we applied it to the explanations generated for each of the 12 charts (each had between ten and twenty explanations). We compared the results of this *color clustering (most-representative)* condition against two other conditions — *color clustering (worker average)* which used the individual color clustering results from all workers, and an *unclustered* condition with no redundancy-detection. As a baseline, we also computed "best possible" results for both *color clustering* and *distributed comparison* which simulate the best case result that could possibly be extracted from workers' responses.

Because clustering is subjective and no objective "best" clustering exists, as a ground truth we compared the results against manual clusterings generated by the three expert raters. These experts (all of whom are visualization researchers and authors on this paper) each independently examined all of the explanations and manually clustered them. The experts were given an unlimited amount of time and endeavored to cluster as consistently and objectively as possible.

For the two *color clustering* conditions, we asked ten different workers to cluster the complete set of explanations for each of the 12 charts. We paid workers $0.20 for each task. To prevent workers from gaming the task, we included gold standard explanations. In each task we added two explanations that we knew to be redundant and a third which we knew to be unique. We eliminated workers who failed to group the known redundant explanations together or who grouped the unique explanation with any other response. A total of 91 workers participated, producing 120 total clusterings. Note that some workers performed the task for more than one chart.

In the *color clustering (worker average)* condition, we retained all of the clusterings generated by individual workers, while in the *color clustering (most-representative)* we selected the single most-representative clustering for each chart using the algorithm described in Section 4.2.2. For comparison, we also computed the *color clustering (best possible)* result, which we obtained by selecting the single clustering for each chart that best matched the experts.

As another point of comparison, we also asked a second set of workers to cluster explanations using a *distributed comparison* interface. In this condition, we created a comparison task for every pair of explanations for each chart (1,064 comparison tasks in total). We grouped tasks into batches of 20 and asked five unique workers to complete each batch. Again, we paid workers $0.20 for each batch, and included the same gold-standard checks as in the *color clustering* conditions. A total of 96 workers produced 5,032 comparisons. We then averaged all five workers' scores for each comparison and used k-means clustering to produce a final set of clusters. Because choosing an appropriate number of clusters (k) remains difficult, we report only the *distributed comparison (best possible)* result. To compute the best possible result we used the scores produced by workers to cluster each set of explanations using all possible values of k ($k = 1, 2, ...n$, where n is the total number of explanations for the chart). We then selected the clustering for each chart that most closely matched the experts. This simulates the best result that could theoretically be achieved given an ideal method of selecting k.

5.1.1 Results

We hypothesized that the crowdsourced *color clustering (most-representative)* approach would produce clusterings that were closer to those produced by the experts than those from the default *color clustering (worker average)* or *unclustered* conditions.

We compare clusterings against the expert clusterings using the F-measure, a symmetric similarity metric that is tolerant to small errors on large clusters, but intolerant to bi-directional impurities [2]. The F-measure similarity for two clusterings is reported on a range from 0 to 1, where 1 indicates that the clusterings are identical and 0 indicates that they are completely dissimilar. We selected the F-measure over other common similarity metrics like Cohen's kappa since it better handles cases like ours where the number of clusters is variable and the clusters are not labeled. We scored each clustering by computing

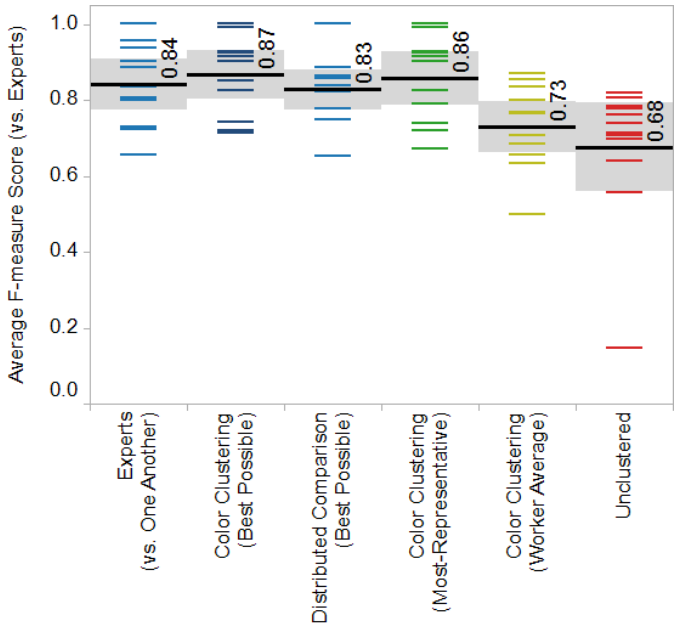

Fig. 7. Results for each of our clustering selection methods. Each column shows the average F-measure similarity between the experts' clusterings and the clusterings produced by the given clustering method. Within a column, colored lines represent the F-measure score of the clustered set of explanations for each individual chart. Black lines and grey bars give the mean and 95% confidence interval for each method. For context, inter-expert results and the results from the *color clustering (best possible)* and *distributed comparison (best possible)* conditions are also included.

the F-measure between it and each of the three expert clusterings individually, then averaging the three results. Results for each condition are shown in Figure 7.

To calibrate our expectations, we compared the three *experts clusterings* against one another. On average, we found that their clusterings were quite consistent ($F = 0.84$). Pairwise comparisons between the individual experts ($E_1 - E_2$: $F = 0.84$, $E_1 - E_3$: $F = 0.85$, $E_2 - E_3$: $F = 0.83$) revealed that no one expert was an outlier.

An ANOVA showed a significant effect for clustering method (*unclustered*, *color clustering (worker average)*, or *color clustering (most-representative)*) on the average F-measure score ($F_{2,33} = 5.55$, $p < .01$). Pairwise t-tests also showed that *color clustering (most-representative)* produced results that were significantly closer to the experts than the *color clustering (worker average)* condition ($p < .01$) or the results from the *unclustered* ($p < .01$) condition, confirming our hypothesis. The difference between the *color clustering (worker average)* and the *unclustered* conditions was not significant.

On average, the *unclustered* results were the least similar to the experts (average $F = 0.68$). This value is non-zero because even the clusters of explanations generated by experts often contain a number of singletons—explanations that do not cluster with any other. As a result, even an unclustered set gets the clustering right for these clusters of size one. The average clusterings produced by workers in the *color clustering (worker average)* condition were somewhat better (average $F = 0.73$). The *color clustering (most-representative)*, approach, however, produced better results across all 12 of our charts (average $F = 0.86$). For almost every chart, the most-representative selection algorithm chose the worker clustering that was the best possible match to the three experts. Moreover, the most-representative clustering was closer, on average, to all three of the experts than the three experts were to one another (average inter-expert $F = 0.84$) and was on-par with both the *distributed comparison (best possible)* ($F = 0.83$) and *color clustering (best possible)* ($F = 0.87$) results. These findings suggest that choosing the most-representative color clustering generates high-quality, internally-consistent clusterings, at least for small sets of explanations.

5.2 Copying and Paraphrasing

We also evaluated how well workers were able to identify paraphrasing from sources. To establish a baseline for how often workers' explanations are copied or paraphrased from the sources they cited, two of our three expert raters examined a sample containing 70 explanations. The two experts individually examined each explanation and the source it cited and coded the explanation as either "copied or paraphrased from the cited source" or "not copied or paraphrased from the cited source". Afterward, the two experts worked together to resolve any differences, and produced a single gold standard. Of the 70 explanations, the experts marked 60% as copied or paraphrased from the source.

We then conducted an experiment to determine how reliably workers could detect paraphrasing. We randomly sampled 20 explanations of the explanations scored by the experts and presented each as a *source-checking microtask* to the crowd. Five crowd workers examined each explanation and source and voted whether the page was or was not "copied or paraphrased from the source". We then tallied these votes and assigned the winning label to each explanation.

The workers' final result matched the experts' for 75% of the explanations. All of the incorrect cases we observed were false negatives—workers indicated that results were not drawn from the source, while the experts deemed that they were paraphrased. The high number of false negatives suggests that workers as a whole used a more conservative definition of paraphrasing than the experts.

6 The Explanation Management Interface

Once workers have rated and clustered a set of explanations, we must surface that information in a way that allows the analyst to quickly browse the explanations and assess them. To this end, we developed an explanation-management interface (Figure 8 and 9) that provides a number of tools and visual cues intended to help analysts quickly find unique explanations and judge their plausibility. We tailored the interface based on the criteria (C_1 through C_3), described earlier.

Analysts can use this interface to browse, filter, and organize explanations generated by workers. Using the explanation-management tools, they no longer need to read through each and every explanation in order. Instead, they can explore clustered results, filter them by quality and frequency, and get a sense of their provenance.

By default, the interface displays a list of explanations grouped first by chart view and then by cluster. Clusters are initially collapsed, so that only the explanation in the cluster with the highest quality score is visible. The clusters are also sorted based on their quality scores, so that the clusters containing the clearest, most plausible explanations are shown first. The analyst can expand clusters to inspect their individual members, and can filter the set of clusters based on a variety of attributes. In many cases, the analyst may wish to save interesting explanations to a "shoebox" [10] in order to revisit them later in the sensemaking process. Our interface allows analysts to save good explanations or groups by dragging them to a shoebox panel at the right (Figure 8D).

Each cluster in the interface includes a set of visual indicators designed to allow the analyst to quickly make judgements about the explanations it contains, often without reading them. These include indicators of explanation quality and frequency (e.g., cluster size) as well as tools that allow analysts to quickly assess explanation provenance.

6.1 Surfacing Explanation Clarity and Specificity

The interface displays the average quality scores generated by workers in *rating microtasks*. We display the quality score in the upper right corner of each explanation (Figure 9G) and color the score using a red-yellow-green color scale. These quality indicators allow an analyst to quickly determine which explanations are more likely to be clear and specific (criteria C_1). Analysts can also reduce the number of visible explanations by using the filtering controls at the top of the interface to hide explanations and clusters that do not contain explanations with high quality scores.

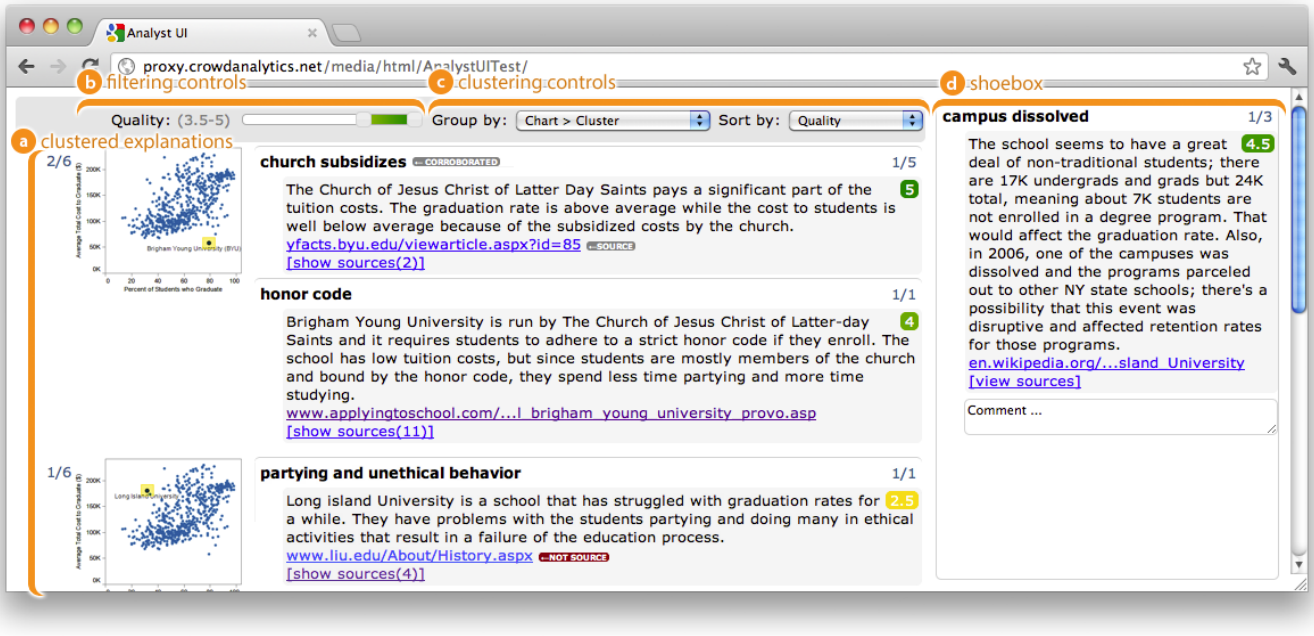

Fig. 8. The explanation-management interface. Explanations (A) can be clustered and collapsed by chart, topic, and source. Filtering (B) and clustering (C) controls allow the analyst to hide low-scoring clusters and control how they are nested. Explanations, clusters, and charts, can be dragged to the shoebox (D) and annotated for later review.

6.2 Surfacing Explanation Redundancy

By default, the system collapses clusters of redundant explanations so that each cluster displays just the highest-quality version of the explanation. Each cluster also contains a count showing the total number of explanations in the cluster and how many are currently visible (Figure 9D). The highest-quality explanation serves as a summary of the cluster and reduces the amount of effort an analyst must expend to examine the explanation. An analyst can also use the cluster size to gauge the frequency and level of support for the explanation (criteria C_2). If the analyst wants to inspect other versions of the explanation, they can expand a collapsed cluster by clicking on the cluster size indicator. Clicking on the indicator a second time re-collapses the cluster.

6.3 Surfacing Explanation Provenance

Each explanation also displays an abbreviated link to any web pages it cites (Figure 9F). These short links allow the analyst to quickly determine if the explanation is drawn from a source that they trust. The analyst can also click the link to view the source page along with any sections of the page highlighted by the worker (criteria $C_{3.1}$).

If an explanation is of particular interest to the analyst, he or she can expose additional provenance information by clicking the "view sources" link on the comment. Clicking the link exposes the complete set of web pages the worker visited while generating the explanation along with detailed timing information. The analyst can use this list to locate and inspect other sources that informed the explanation and help build an understanding of how a worker came to a conclusion (criteria $C_{3.2}$).

If the analyst determines that a specific domain or web page is a good source, he or she may wish to directly explore other explanations that are drawn from that source. In our own experience, the sources which provide the best explanation for one chart may also provide good explanations for others (for example pages from the Bureau of Labor Statistics provide good explanations for changes in employment in many different US cities). Therefore, our interface also allows the analyst to group explanations based on the sources they cite to quickly find multiple explanations drawn from the high-quality sources.

6.4 Surfacing Paraphrasing and Worker Additions

In the explanation-management interface, we provide a provenance indicator next to the source URL (Figure 9E) of each explanation that *source-checking* workers identified as a copy or paraphrase. Based on our experiments, we place indicators next to explanations that were identified either as "paraphrased" or "not related" by more than 50% of workers. This indicator allows analysts to quickly identify explanations that are drawn directly from a source before reading them. Knowing an explanation was copied or paraphrased from a known source can allow an analyst to make confidence judgments based on that source's reputation. High-quality paraphrased explanations also serve as leads to help analysts identify good web resources that they may wish to utilize directly.

6.5 Surfacing Corroborating Explanations

An explanation that cites multiple reliable sources is more likely to be credible than one that cites only a single reliable source (criteria $C_{3.3}$). Therefore an analyst may wish to know if multiple versions of an explanation in a cluster cite the same source or refer to multiple independent ones. In our interface, workers can assess this directly by expanding a cluster and grouping the responses within in by URL or domain. We also provide a "multiple sources" indicator in the heading of clusters that contain corroborating citations. Mousing over this indicator displays a list of sources along with the number of explanations in the group that cite them. This indicator serves as a shortcut for analysts, allowing them to quickly make confidence judgments based on corroborating sources without examining explanations or sources individually.

7 Discussion and Future Work

Here we offer a few observations based on our experience collecting, clustering, and exploring crowdsourced explanations.

7.1 Explanation Segmentation

Our current implementation asks workers to separate distinct explanations into separate fields in the explanation microtask and allows them to select different source text for each. However, in practice, many workers still give multiple candidate explanations as part of a

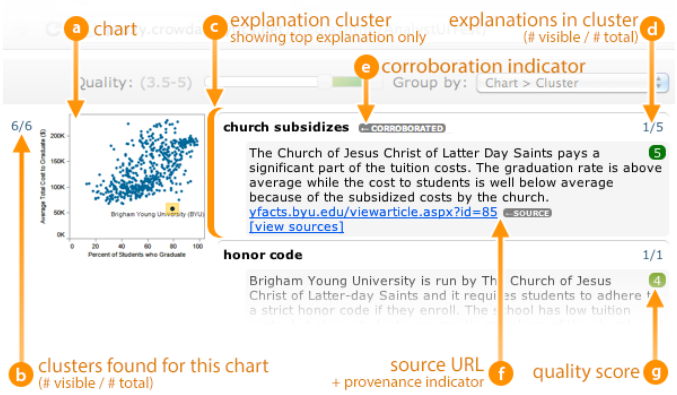

Fig. 9. A closeup of the explanation-management interface, highlighting a single chart (A) with two explanation clusters. Each chart includes an indicator (B), showing the number clusters of explanations. Each cluster (C) displays a count showing explanations it contains (D) and an indicator showing if the explanation is corroborated by multiple sources (E). Each individual comment displays a source URL and provenance indicator (F) along with a color-coded quality score (G).

single paragraph or sentence. This mixing of explanations can make responses difficult to group, since a single response may contain two or more ideas that belong in disparate clusters.

One possible solution to this issue is to modify the explanation-generation tasks to force workers to produce more cleanly segmented explanations. However, such segmentation can be difficult to enforce, especially when explanations are interrelated. Alternately, workers in intermediate *segmentation microtasks* could break apart compound responses into their constituent explanations, but this introduces the potential for information or intent to be lost as workers break apart or alter others' explanations.

This experience speaks to the broader issue of task granularity when crowdsourcing open-ended tasks. Breaking tasks into small, modular components makes it easier to compose tasks together and process results systematically. Small, straightforward tasks also reduce the potential for worker error, and make it easier to identify and discard poor results. However, small, segmented tasks may inhibit contributions from talented or knowledgable workers, since they are not free to explore, author, or contribute outside the constraints of the task and cannot bring their expertise to bear on areas of the problem where it might be beneficial.

7.2 Crowd Composition

Our approach assumes a crowd composed largely of non-expert workers whose responses may be of variable quality—for example, workers recruited in online task markets like Mechanical Turk. With these workers in mind, we designed the analysis microtasks to be simple and include little interactivity. We also include quality-rating and source-checking tasks analysis microtasks to help filter out low quality results and help analysts identify the most likely explanations.

However, more complex analyses or datasets that require specific domain knowledge or contain sensitive information may call for the use of more knowledgeable (and potentially private) crowds. We believe analysts could utilize a workflow similar to ours to systematically collect and integrate findings from large crowds of trusted workers. Using a crowd of trusted workers, some quality-control mechanisms could be relaxed, reducing the number of post-processing steps and giving workers more freedom to explore. For example, trusted workers could be given the freedom to manipulate the visualization and explore alternate views of the dataset that might inform their explanations. Trusted workers could also self-assess the quality of their explanations and sources, reducing the number of steps in workflow while still providing metadata that analysts can use to filter and reorganize their results.

8 Conclusion

In this paper, we have shown how crowdsourcing tools can help analysts explore datasets and identify good possible explanations for trends and patterns in their data. Specifically, we demonstrate that crowd workers can assist analysts, not only by visually examining and explaining datasets but also by helping organize and filter those explanations. Tools like these, which allow analysts to enlist greater numbers of outside collaborators will be increasingly important as analysts seek to make sense of larger and more diverse datasets. By exploring how crowd workers can compliment analysts' effort, both by analyzing data directly and by carrying out other supporting tasks, this work highlights the potential of crowd-assisted analysis tools to come.

References

[1] Same Origin Policy. http://www.w3.org/Security/wiki/Same_Origin_Policy.

[2] Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval 12*, 4 (July 2008), 461–486.

[3] Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of UIST*, ACM (2010), 313–322.

[4] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. VizWiz: nearly real-time answers to visual questions. In *Proc. of UIST* (2010), 333–342.

[5] Chilton, L., Little, G., Edge, D., Weld, D. S., and Landay, J. A. Cascade: Crowdsourcing Taxonomy Creation. In *Proc. of CHI*, ACM (2013).

[6] Fisher, K., Counts, S., and Kittur, A. Distributed sensemaking: improving sensemaking by leveraging the efforts of previous users. In *Proceedings of CHI*, ACM (2012), 247–256.

[7] Heer, J., Viégas, F. B., and Wattenberg, M. Voyagers and voyeurs: Supporting asynchronous collaborative visualization. *Communications of the ACM 52*, 1 (2009), 87–97.

[8] Luther, K., Counts, S., Stecher, K. B., Hoff, A., and Johns, P. Pathfinder: An Online Collaboration Environment for Citizen Scientists. In *Proceedings of CHI*, ACM (2009), 239–248.

[9] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] Pirolli, P., and Card, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis 5* (2005).

[11] Rzeszotarski, J. M., and Kittur, A. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of UIST*, ACM (2011), 13–22.

[12] Stone, B., Dennis, S., and Kwantes, P. J. Comparing Methods for Single Paragraph Similarity Analysis. *Topics in Cog. Sci. 3*, 1 (2010), 92–122.

[13] Surowiecki, J. *The Wisdom of Crowds*. Anchor, 2005.

[14] Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. T. Adaptively Learning the Crowd Kernel. *arXiv.org* (2011).

[15] Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*, 2 (2001), 411–423.

[16] Trushkowsky, B., Kraska, T., Franklin, M. J., and Sarkar, P. Getting It All from the Crowd. *arXiv.org* (2012).

[17] Viégas, F. B., Wattenberg, M., Ham, F. v., Kriss, J., and McKeon, M. Many Eyes: A Site for Visualization at Internet Scale . *Trasactions on Visualization and Computer Graphics 13*, 6 (2007), 1121–1128.

[18] Viégas, F. B., Wattenberg, M., McKeon, M., Ham, F. V., and Kriss, J. Harry potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. In *Proceedings of HICSS* (2008).

[19] Willett, W., Heer, J., and Agrawala, M. Strategies for crowdsourcing social data analysis. In *Proceedings of CHI*, ACM (2012), 227–236.

[20] Willett, W., Heer, J., Hellerstein, J. M., and Agrawala, M. CommentSpace: Structured Support for Collaborative Visual Analysis. In *Proceedings of CHI*, ACM (2011).

[21] Wu, M., and Marian, A. Corroborating answers from multiple web sources. In *Proceedings of WebDB* (2007).

[22] Yi, J., Jin, R., Jain, A. K., and Jain, S. Crowdclustering with Sparse Pairwise Labels: A Matrix Completion Approach. In *Workshops at the Conference on Artificial Intelligence*, AAAI (2012).

[23] Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., and Horvitz, E. Human Computation Tasks with Global Constraints. In *Proceedings of CHI*, ACM (2012), 217–226.