Extracting References Between Text and Charts via Crowdsourcing

Nicholas Kong, Marti A. Hearst, Maneesh Agrawala
University of California, Berkeley
{nkong,hearst,maneesh}@berkeley.edu

ABSTRACT
News articles, reports, blog posts and academic papers often include graphical charts that serve to visually reinforce arguments presented in the text. To help readers better understand the relation between the text and the chart, we present a crowdsourcing pipeline to extract the references between them. Specifically, we give crowd workers paragraph-chart pairs and ask them to select text phrases as well as the corresponding visual marks in the chart. We then apply automated clustering and merging techniques to unify the references generated by multiple workers into a single set. Comparing the crowdsourced references to a set of gold standard references using a distance measure based on the F_1 score, we find that the average distance between the raw set of references produced by a single worker and the gold standard is 0.54 (out of a max of 1.0). When we apply clustering and merging techniques the average distance between the unified set of references and the gold standard reduces to 0.39; an improvement of 27%. We conclude with an interactive document viewing application that uses the extracted references; readers can select phrases in the text and the system highlights the related marks in the chart.

Author Keywords
visualization; crowdsourcing; interactive documents

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces

INTRODUCTION
Charts abound in news articles, reports, blog posts, and academic papers. They are widely used to present large amounts of data, illustrate trends or differences, and emphasize key points presented in the text [7]. Charts can also provide additional data, not mentioned in the text, to give readers more context and allow them to make their own inferences. Thus, for readers to fully understand such a document they must parse all of the *references* between the text and the corresponding visual marks (e.g., bars, lines, points, pie slices, etc.) in the charts.

Yet, identifying such references between the text and the chart can be challenging. Consider the example in Figure 1 from a Pew Research report [28]. The text explains that *"Half or more in 13 of 21 nations surveyed believe that most people can succeed if they are willing to work hard."* To find the corresponding nations in the chart, the reader must identify the countries for which the orange bar (most succeed if work hard) is longer than the blue bar (hard work no guarantee). Often, as in this case, the text only refers to a subset of the data in the chart and the reader must perform complex visual comparisons to identify the correct subset. In other cases, the text may paraphrase values in the chart and require the reader to bring external information to bear. For example, the text may use the term "EU" to refer to a subset of the European countries in the chart.

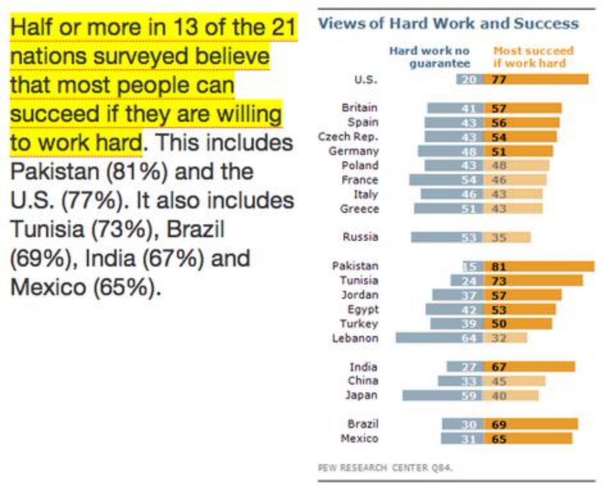

Half or more in 13 of the 21 nations surveyed believe that most people can succeed if they are willing to work hard. This includes Pakistan (81%) and the U.S. (77%). It also includes Tunisia (73%), Brazil (69%), India (67%) and Mexico (65%).

Figure 1. Example of a reference between the text and a chart from a Pew Research report [28]. The highlighted text (yellow background) refers to the 13 bar segments highlighted in the chart (saturated orange). We use our crowdsourcing pipeline to generate these references and display them here in our interactive document viewing application.

We present a crowdsourcing pipeline that takes a document containing text and one or more charts as input, and extracts the references between the text and the chart. In the preprocessing stage of our pipeline we use a mix of manual and algorithmic techniques [29] to segment the document into paragraph-chart pairs and then extract the marks and data from each chart. In the reference extraction stage, we give crowd workers paragraph-chart pairs and ask them to select text phrases as well as the corresponding visual marks in the chart. We then apply automated clustering and merging techniques to unify the references generated by multiple workers into a single set of references.

We compare the crowdsourced references to a set of gold standard references created by two experts using a distance measure based on the F_1 score [24]. We find that the average distance between the raw set of references produced by a single worker and the corresponding gold standard references is

0.54 (out of a max of 1.0). When we apply our clustering and merging techniques the average distance between the unified set of references and the gold standard reduces to 0.39; an improvement of 27%.

Finally, as a proof-of-concept application, we present an interactive document viewer that uses the extracted references to improve the reading experience. As shown in Figure 1 readers can select phrases in the text and our application automatically highlights the related marks in the chart. This application demonstrates how access to the references can help readers better understand the relationship between the text and charts in a document.

RELATED WORK

Our work builds on three main areas of related work.

Crowdsourcing for data collection

Crowdsourcing has become a popular way to collect data in a variety of disciplines. We consider prior work from both the visualization and natural language processing (NLP) communities in the design of our pipeline. In the context of visualization, researchers have used crowdsourcing to gather data for graphical perception experiments [12, 14, 20], and to perform data analysis on charts [33]. In the NLP community, Snow et al. [31] have demonstrated the viability of crowdsourcing to cheaply generate large sets of labeled training examples for a variety of text analysis tasks. Others have used it to create training data for sentiment analysis [13], identify entities across languages [25], and find paraphrases for noun compounds [26]. Callison-Burch and Dredze [6] provide a recent survey of uses of crowdsourcing in NLP research. Our work uses crowdsourcing to address a new problem; extracting references between the text and charts within a document.

Crowdsourcing pipelines for complex tasks

There has been much recent work on designing crowdsourcing pipelines for complex tasks including taxonomy creation [8], explaining outliers and trends in data analysis [33], and generating answers to uncommon queries [3]. Others have designed toolkits such as TurKit [23], CrowdForge [18], Turkomatic [21], and Jabberwocky [1], to help developers implement complex crowdsourced pipelines. Our crowdsourcing pipeline for extracting references draws inspiration from the designs of these other pipelines and toolkits. We focus on developing algorithmic techniques for combining the references extracted by multiple workers into a unified set.

Annotating Visualizations

Visualization researchers have recognized that the most effective charts use labels, captions and other text annotations to clarify and accentuate important points [9]. Such text annotations can direct the readers attention through a visualization [30], provide additional semantics and meaning for the data [5], and emphasize specific interpretations [15]. Researchers have developed automated techniques for adding text annotations to a chart in order to draw attention to outliers and trends in the data [17] or to provide additional context for the stock chart data [16]. However, these either assume the text used for annotation resides in the dataset (e.g., as metadata) or that it can be obtained through domain-specific searches (e.g., searching a news database using a stock ticker symbol). Our crowdsourcing pipeline for extracting references between text and charts is premised on the idea that charts often appear within documents that contain additional text explaining the chart. Our interactive document viewing application is designed to highlight such references and thereby annotate the chart with explanatory text.

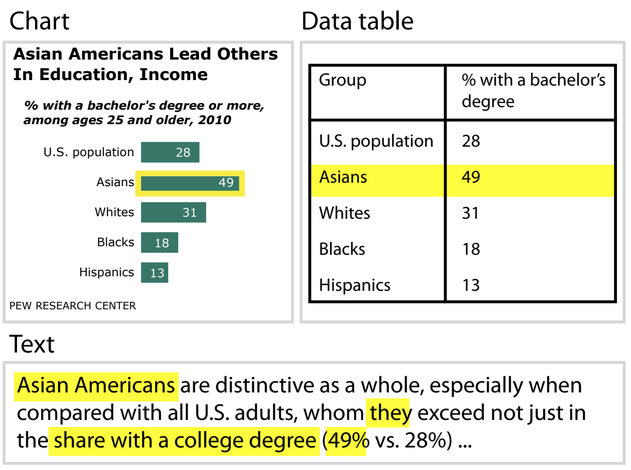


Chart
Asian Americans Lead Others In Education, Income
% with a bachelor's degree or more, among ages 25 and older, 2010
U.S. population 28
Asians 49
Whites 31
Blacks 18
Hispanics 13
PEW RESEARCH CENTER
Data table
Text

Group	% with a bachelor's degree
U.S. population	28
Asians	49
Whites	31
Blacks	18
Hispanics	13

Asian Americans are distinctive as a whole, especially when compared with all U.S. adults, whom they exceed not just in the share with a college degree (49% vs. 28%) ...

Figure 2. Text phrases (bottom) and visual marks in a chart (top-left) refer to data tuples (or rows) of an underlying data table (top-right). To extract the references we must identify the relationships between the text phrases, the visual marks and the data tuples.

REFERENCE EXTRACTION PIPELINE

In a document containing text and charts, there are two kinds of references: (1) text phrases may refer to data and (2) visual marks may refer to data (Figure 2). The referent in both cases is one or more tuples (or rows) of an underlying data table. To solve the reference extraction problem we must recover both of these types of references for the input document. In this work we focus primarily on the first goal and rely on a combination of prior methods [29] and manual techniques to achieve the second goal. Throughout the paper, we will use the term *reference* to mean either the correspondence between a set of phrases (in the text) and data tuples or a set of visual marks (in the chart) and data tuples.

Our reference extraction pipeline (Figure 3) takes a document containing text and charts as input, and outputs the references between the text and charts. It consists of three main stages: a *pre-processing* stage in which we set up the crowdsourcing task; a *crowdsourcing* stage, in which we ask a group of workers to extract the references; and a *clustering and merging stage*, in which we combine the worker generated references into a unified reference set.

STAGE 1: PRE-PROCESSING

The pre-processing stage sets up the crowdsourcing task by segmenting input documents into paragraph-chart pairs and extracting the marks and data table from the charts.

Segmenting the document into paragraph-chart pairs

One of the challenges in crowdsourcing is to design relatively small microtasks that workers can complete quickly in good faith [18]. Because reading a long, multi-paragraph document can be time consuming, we design the crowdsourcing

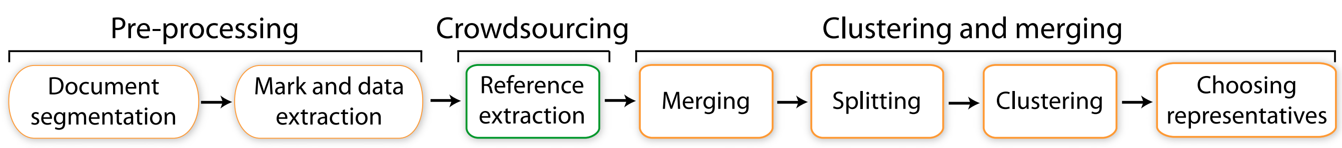


Figure 3. The three stages in our proposed pipeline for extracting references. Orange components are algorithmic or manual, while green components are crowdsourced. Given a document (left), we segment the paragraphs and charts and extract the marks and data in a preprocessing stage. We then ask workers to extract the references in the crowdsourcing stage. Finally, we cluster and merge the worker references into a unified set of references.

task so that each crowd worker only extracts references for a paragraph-chart pair (see next section on Stage 2 of our pipeline). Thus, in the pre-processing stage we split the input document into paragraphs and then manually pair each paragraph with the charts that are related to it (i.e., the text in the paragraph refers to the chart).

While we perform the pairing manually, this task is likely to be amenable to crowdsourcing. For example we could ask crowd workers to mark paragraphs and charts that are related. It may also be possible to automatically compute the pairing by analyzing the document layout as well as standard text references to figures (e.g., "see Figure 2"). However, in this work we focus on the key problem of extracting the references between text phrases and data tuples rather than on pairing the paragraphs with the charts. We leave it to future work to crowdsource or automate the pairing problem.

Mark and data extraction

To extract references between the text surrounding a chart and the data tuples encoded in the chart, we must first recover the data table from the chart. We assume the charts in the input document are bitmaps; thus, we recover the data table in two steps. We first analyze the chart to identify the locations of all of the data-encoding marks (e.g., bars in a bar chart) as well as the chart axes. We then analyze the marks themselves in relation to the axes to recover the underlying data values.

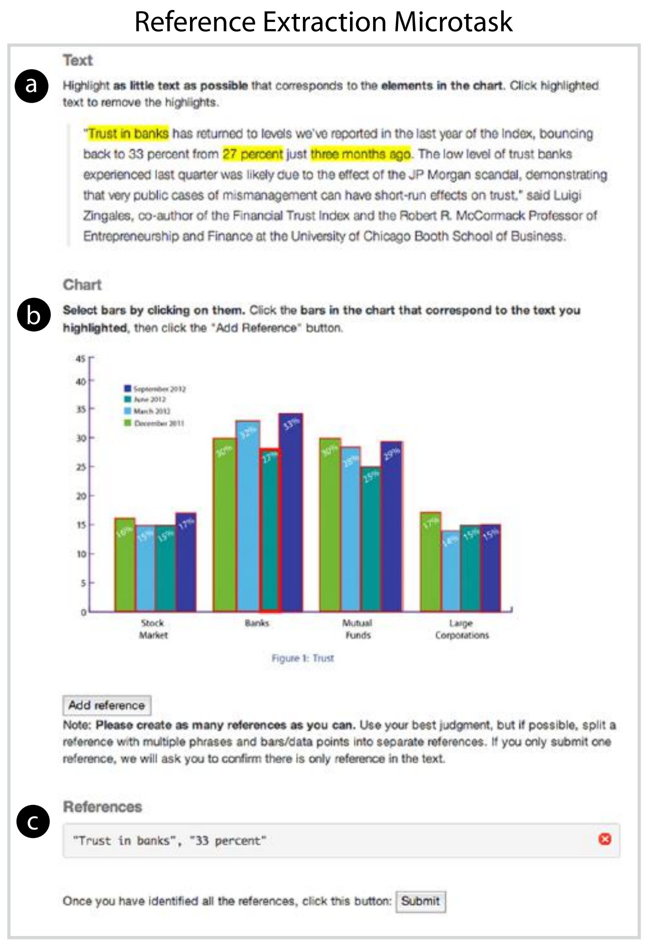
We rely on ReVision [29] to automatically extract the marks, axes and data values from simple bar charts. For more complex charts (e.g., stacked or grouped bar charts), we built a simple graphical interface for interactively annotating the marks and associating data values with them. We also use this tool to correct errors or missing output from ReVision.

STAGE 2: CROWDSOURCING REFERENCE EXTRACTION

In the crowdsourcing stage of the pipeline, we ask workers to mark text phrases and the corresponding data-encoding marks for a set of paragraph-chart pairs. To maintain high-quality work we train the workers and regularly check the accuracy of their references on a small set of *gold tasks*.

Reference extraction microtask

Figure 4 shows our microtask interface for extracting references between the text and data. The microtask includes a paragraph of text and a chart. We ask workers to select one or more text phrases (highlighted in yellow) and click on the corresponding set of marks in the chart (highlighted with red outlines). Since the pre-processing stage gives us the mapping between marks and data tuples we can link the worker selected text phrases to the relevant data tuples and thereby form the complete reference between text and data.


Reference Extraction Microtask

Text

a Highlight **as little text as possible** that corresponds to the **elements in the chart**. Click highlighted text to remove the highlights.

"Trust in banks has returned to levels we've reported in the last year of the Index, bouncing back to 33 percent from 27 percent just three months ago. The low level of trust banks experienced last quarter was likely due to the effect of the JP Morgan scandal, demonstrating that very public cases of mismanagement can have short-run effects on trust." said Luigi Zingales, co-author of the Financial Trust Index and the Robert R. McCormack Professor of Entrepreneurship and Finance at the University of Chicago Booth School of Business.

Chart

b **Select bars by clicking on them.** Click the **bars in the chart that correspond to the text you highlighted**, then click the "Add Reference" button.

Note: **Please create as many references as you can.** Use your best judgment, but if possible, split a reference with multiple phrases and bars/data points into separate references. If you only submit one reference, we will ask you to confirm there is only reference in the text.

c References

"Trust in banks", "33 percent"

Once you have identified all the references, click this button: Submit

Figure 4. The reference extraction microtask presents a paragraph (a) and a chart (b). Workers must select text phrases (highlighted in yellow) and clock on the corresponding visual marks in the chart (thick red outline). The thin red outlines in the chart indicate selectable marks. Clicking "Add reference" adds a row to the list at the bottom of the chart (c) showing the text of the reference. This list holds all of the references the worker has already created for this paragraph-chart pair.

Clicking the "Add Reference" button adds the reference to a list shown at the bottom of the task (Figure 4c). Workers can remove references from the list by clicking the **x** button and can add as many references as they wish before clicking the "Submit" button to finish the task. Thus, the worker can create one or more references for each paragraph-chart pair and each reference relates a set of words in the paragraph to a set of data tuples.

Quality control

Crowdworkers do not always produce accurate, high-quality work. They may not understand the task, they may be lazy and do as little work as possible, or they may maliciously introduce errors [2]. We have designed our microtasks to control for work quality in two ways: (1) we require workers to pass a training task before they can submit references, and (2) we intersperse the regular tasks with *gold tasks* where the correct answer is known a priori [22, 27] to check that a worker completed the task correctly. In this section we assume the existence of paragraph-chart pairs for which we have a gold standard set of references as well as a quantitative distance measure for comparing worker generated references to the gold standard. We will describe how we created the gold standard references and the distance measure in the following section.

Training workers

First-time workers must pass a training task before they can submit work for the pipeline. The training task describes the reference extraction task and presents a paragraph-chart pair with a corresponding set of gold references. It then gives the workers another paragraph-chart pair for which the gold standard references are known but this time asks them to extract the references themselves. We check for correctness by comparing the worker generated references to the gold standard references using our distance measure (see section on Reference Comparison). If the distance is high (> 0.5 out of a max of 1.0), we ask the worker to extract the references again and resubmit. Workers may submit references as many times as they wish but can only move on to the main task once they have completed the training task accurately.

Interspersed gold tasks

Workers may perform well on our training task but still perform poorly on other, more difficult tasks. They may also complete later tasks in bad faith. We therefore continue to monitor worker performance by asking them to complete two reference extraction tasks within each microtask: one for which we have gold standard references and one for which we do not. We randomize the ordering of the gold and non-gold tasks. If the worker correctly extracts the references for the gold standard task (i.e. the distance between the worker reference set and the gold reference set < 0.42), we keep their result for the non-gold; otherwise, we discard their result. In the event that a worker has already completed all of the available gold tasks, we only show the worker the non-gold task, and we use their previous performance on the gold tasks to decide whether to keep or discard their result.

GOLD REFERENCES AND REFERENCE COMPARISON

To assess the quality of worker generated references, we created gold standard references for a set of paragraph-chart pairs. We also designed two distance measures for comparing worker generated references to the gold references: a distance between a *single* worker reference and a *single* gold reference, and a distance between a *set* of worker references and a *set* of gold references.

Creating gold standard references

In any paragraph-chart pair only a subset of the text phrases refer to data. We designed gold standard references to capture the minimal set of text phrases and data tuples that uniquely correspond to one another. An example of a *minimal reference* is shown in Figure 2; adding more text could only increase the number of tuples in the reference while removing phrases would make the correspondence with the data tuple ambiguous. Figure 1 illustrates another minimal reference in which the minimal text refers to multiple data tuples.

Two experts (the first author and a post-doc from our lab) created a gold standard set of references using an iterative process. We jointly drafted an initial set of guidelines for creating minimal references, independently extracted minimal references from five paragraph-chart pairs and then jointly revised the guidelines based on the results. We then separately extracted references for the remaining corpus of paragraph-chart pairs (see Results section for description of the corpus). Finally, we came to a consensus on the paragraph-chart pairs where we disagreed on the content or number of references to produce the final set of gold references. Appendix A contains the list of guidelines we used to create minimal references.

Distance measures for comparing references

For each paragraph-chart pair, each worker can submit one or more references linking text phrases to data tuples. To compare the worker generated references to the gold standard references we first compute the distance between each worker generated reference and each gold reference. We then combine these single reference distances to compute the overall distance between the set of worker generated references and the set of gold standard references. In the following discussion, we use lower case letters w and g to refer to a single worker or gold reference respectively. We use upper case letters W and G to refer to sets of references.

Distance: Single worker to single gold reference

A reference is composed of a collection of phrases (or equivalently, a collection of words), and a corresponding collection of data tuples. To compute the distance between two references we separately measure the similarity between the phrases and between the data tuples using the F_1 score [24] and combine the scores as follows

$$d(w,g) = 1 - \left(F_1^{text}(w,g) \cdot F_1^{data}(w,g)\right). \qquad (1)$$

The F_1 score is a common statistical measure of the similarity between two collections [24]. It is based on measures of precision and recall and is computed as follows,

$$\begin{aligned} \text{precision}(w,g) &= \frac{|w \cap g|}{|w|}, & (2)\\ \text{recall}(w,g) &= \frac{|w \cap g|}{|g|}, & (3)\\ F_1(w,g) &= \frac{2 \cdot \text{precision}(w,g) \cdot \text{recall}(w,g)}{\text{precision}(w,g) + \text{recall}(w,g)} & (4)\end{aligned}$$

In these equations we have overloaded the notation; we use w and g to denote either the collection of text phrases or the collection of data tuples for the worker or gold reference respectively. To compute $F_1^{text}(w,g)$ we treat w and g as collections of text phrases and to compute $F_1^{data}(w,g)$ we treat them as collections of data tuples. Our single reference distance measure $d(w,g)$ lies in the range $[0,1]$, where 0 denotes an exact match, and 1 denotes no similarity in either the text phrases or data tuples of the worker and gold reference.

Distance: Set of worker to set of gold references

A worker's set of references W is close to a gold set of references G if each reference in the worker's set has a low distance to the nearest reference in the gold set and vice versa. Figure 5 shows how we compute the distance $d(W, G)$ between these sets of references. For each worker reference we find the nearest gold reference; that is, for each $w \in W$ we find the $g \in G$ such that $d(w, g)$ is minimized. This gives us a correspondence between each worker reference and a gold reference. However, we may be left with unmatched gold references. For each unmatched gold reference $g \in G$ we find the nearest worker reference $w \in W$ by minimizing $d(w, g)$. Finally, we compute $d(W, G)$ as the sum of the distances between corresponding worker and gold references, normalized by the total number of correspondences.

Handling references with extraneous text

As described earlier, we designed our gold standard references to be minimal. Although our reference extraction microtask asks workers to submit such minimal references, in practice we have found that workers often include *extraneous text phrases* that do not have any impact on the relationship between the text and the data tuples. For example, Figure 2 shows a paragraph-chart pair with a minimal set of text phrases corresponding with the tuple (Asians, 49). Suppose a worker marks the following phrases

> *Asian Americans are distinctive as a whole*, *especially ... share with a college degree* (*49%* vs. 28%)

as corresponding with this tuple. In this case the worker's phrases contain all of the text in the minimal reference, but also add the extraneous words "*are distinctive as a whole*". The worker's reference remains unambiguous even though it is not minimal.

Since such extraneous words are common in worker references and do not increase ambiguity we ignore extraneous words in our distance computations. Specifically for each paragraph-chart pair we construct the set of extraneous words by starting with all the words in the paragraph and removing all of the text phrases that appear in the gold standard references. We then remove the extraneous words from each worker reference before computing a distance.

STAGE 3: CLUSTERING AND MERGING REFERENCES

The set of references generated by a single worker for a given paragraph-chart pair can be inaccurate; they may miss references and form an incomplete set, contain incorrect references, or include references that are not minimal. To reduce such problems we ask multiple workers (10 in our experiments) in the crowdsourcing stage to independently generate a set of references for each paragraph-chart pair. In the clustering and merging stage we algorithmically combine the independently generated references to form a set of *unified references*. The goal of this stage is to integrate the most accurate parts of the worker generated references. We perform the following operations:

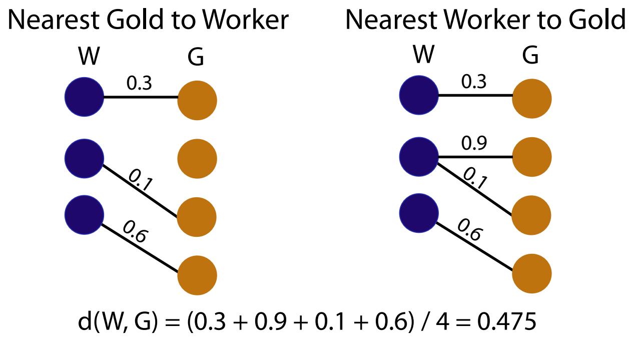


Figure 5. Computing $d(W, G)$ for a set of worker references W (blue) and a set of gold references G (orange). (left) For each worker reference, we find the nearest gold reference. Here, one gold reference remains unmatched. (right) So for each such unmatched gold reference, we find the nearest worker reference. Finally, we compute the distance by averaging the distances between corresponding worker and gold references.

1. Merge references that contain the same set of text phrases but different data tuples into a single reference.
2. Compare references across all workers to identify those that are subsets of one another. Then split the larger references into smaller references using the subset relationship.
3. Cluster all of the resulting references based on similarity.
4. Choose a single representative reference for each cluster resulting to form the output set of *unified references*.

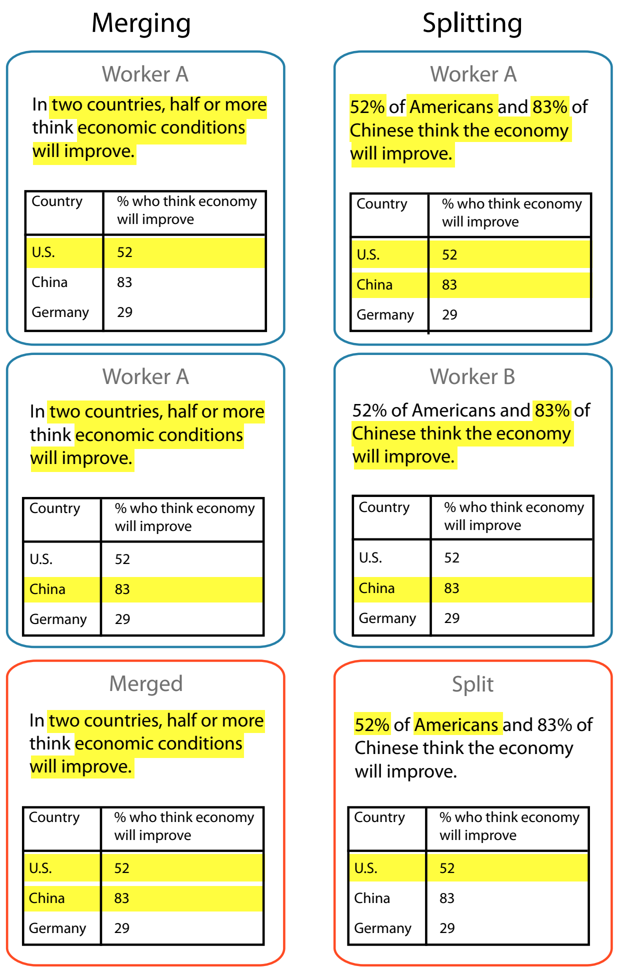
1. Merging references containing the same text phrases

Individual workers sometimes create different references that include the same collection of text phrases but different data tuples. We assume each such reference is incomplete and merge them into a single reference by taking the union of their phrases and data tuples. In practice we have found that we produce more accurate results if we merge references even when there are small differences in their text (i.e., a few extra or missing words). Therefore, we merge references that share 95% of their words. Figure 6 (left) shows an example of two references generated by the same worker that we merge into a single reference using this process. The merged reference contains all of the words and tuples contained in the original worker generated references.

2. Splitting references based on subsets

When comparing references across multiple workers it is common to find references that are subsets of one another. Consider the worker generated references shown Figure 6 (right). Let a_t and a_d denote the collections of text phrases and data tuples respectively for reference a (Figure 6 (right, top)) and let b_t and b_d denote the corresponding collections for reference b (Figure 6 (right, middle)). In this case b is a subset of a because $b_t \subseteq a_t$ and $b_d \subseteq a_d$. That is, the text and data tuples of b are contained in a.

We split the larger reference a into smaller, more minimal references by iteratively subtracting all such subset references b from it. That is, we replace a with $a - b$ by replacing a_t and a_d with $a_t - b_t$ and $a_d - b_d$ respectively. We continue subtracting all subset references b in this manner until either a_t



In two countries, half or more think economic conditions will improve.

Country	% who think economy will improve
U.S.	52
China	83
Germany	29

52% of Americans and 83% of Chinese think the economy will improve.

Country	% who think economy will improve
U.S.	52
China	83
Germany	29

Figure 6. (left) For each worker, we merge references with the same text but different tuples by taking the union of the text and tuples. (right) We split larger references into smaller, more minimal references by finding references that are subsets of one another. Here, Worker B's reference (middle) is a subset of one submitted by Worker A (top). We subtract B's reference from A's to obtain a more minimal reference (bottom).

or a_d is empty or we have subtracted all of the subset references b. The resulting reference a after subtraction is usually much closer to minimal (Figure 6 (right, bottom)). We repeat this subtraction process for every large reference a for which we find subset references b. Note that to properly account for the subtracted references in the fourth step of clustering and merging (choosing representative references) we duplicate each subtracted reference b and add it to the pool of worker generated references.

3. Clustering references by similarity

Next, we cluster the references. Since we ask multiple workers to submit references for each paragraph-chart pair, we expect many references to be similar. To eliminate such redundancy we first cluster the similar references and then choose a representative reference for each cluster. This ensures that the final set of unified references are as distinct as possible.

To cluster the references we first form a graph in which each reference is a node and we create an edge between references a and b if $d(a, b) < 0.32$. This condition ensures that connected references are similar to one another. We chose the 0.32 distance threshold empirically and found it to give good results in practice. To build the clusters we compute maximal cliques for this graph using the following greedy algorithm. We initialize the first clique with the reference containing the most tuples (in case of a tie we pick randomly). We then iterate through the remaining references that haven't been added to a clique, adding a reference to the clique if it is connected to every other reference in the current clique. When it is impossible to add another reference to the clique we start the process again creating a new clique. We repeat this clique-building process until all the references are part of a clique. Finally, we treat each resulting clique as a cluster.

4. Choosing representative references

References within a cluster are guaranteed to be similar to one another but may not be identical. In this final step, we choose a representative reference for each cluster.

Within each cluster we first group together references that share at least 95% of their words and contain exactly the same set of data tuples. The references within each such group are almost identical. We consider the group containing the largest number of references and if the size of this largest group is greater than three we select the reference within it that contains the most words as the representative for the cluster. In this case we are essentially selecting a reference that was submitted by three or more workers as the representative. The references within each group all contain the identical set of data tuples. So if there is a tie in the size of the largest group we pick the representative reference from the group containing the largest number of data tuples. This choice is based on the assumption that the group containing fewer data tuples is incomplete and omits one or more relevant tuples. We have found this tie-breaking procedure to work well in practice.

If no groups within a cluster contain three or more references we create a representative reference r for the entire cluster as follows. We set the collection of text phrases for the representative r_t to include all the words that appear in three or more references in the cluster. Similarly we set the collection of data tuples for the representative r_d to include all the tuples that appear in three or more references in the cluster. Finally, if a cluster contains fewer than three references we create r by taking the union of both the text phrases and data tuples across all of the references in the cluster. Thus, in both cases we synthesize a representative reference that combines information contained in multiple worker generated references.

RESULTS

To test our pipeline we gathered a corpus of 18 documents from the Web containing 35 bar charts. We targeted documents written for a general audience, such as blogs on news websites (the Economist's Graphic Detail [10], the Guardian's DataBlog [11]) and reports from agencies focusing on public policy (Pew Research [28], and a governmental health services agency [32]). We used ReVision to extract the marks and data from 20 of the charts and manually extracted the marks and data from the rest. We then manually

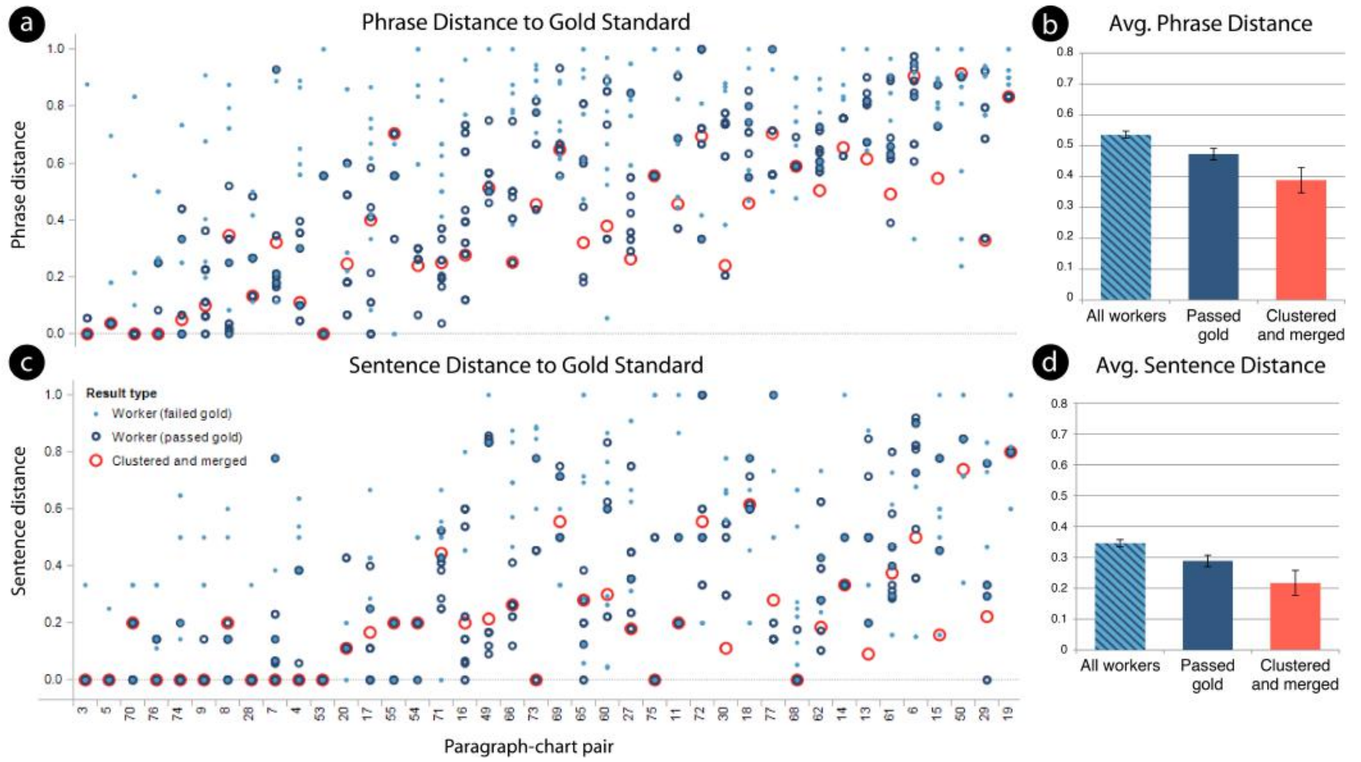


Figure 7. (a) Phrase distances between the worker-generated references (blue), clustered and merged references (red) and the gold standard references (lower is better). Each column represents a paragraph-chart pair. (b) Average phrase distance aggregated across all 40 paragraph-chart pair conditions. (c) Sentence distances between worker-generated references and the gold standard references. (d) Average sentence distances.

split the documents into 49 paragraph-chart pairs. We withheld 9 of these pairs for use in gold tasks and gathered worker references for the remaining 40. We also asked our experts to manually produce gold standard reference sets for all 49 pairs using the procedure described earlier.

We then used Amazon's Mechanical Turk to ask 10 crowd workers to extract references for each of the 40 paragraph-chart pairs, yielding a total of 400 reference sets generated by 77 unique crowd workers. Each microtask asked workers to extract references for one paragraph-chart pair and paid \$0.15, for a total cost of \$60. We deployed multiple small batches of tasks at different times of day and on different days of the week. Overall, it took about 7 days and 16 hours to gather all 400 worker generated reference sets.

Merging and clustering improves on the average worker

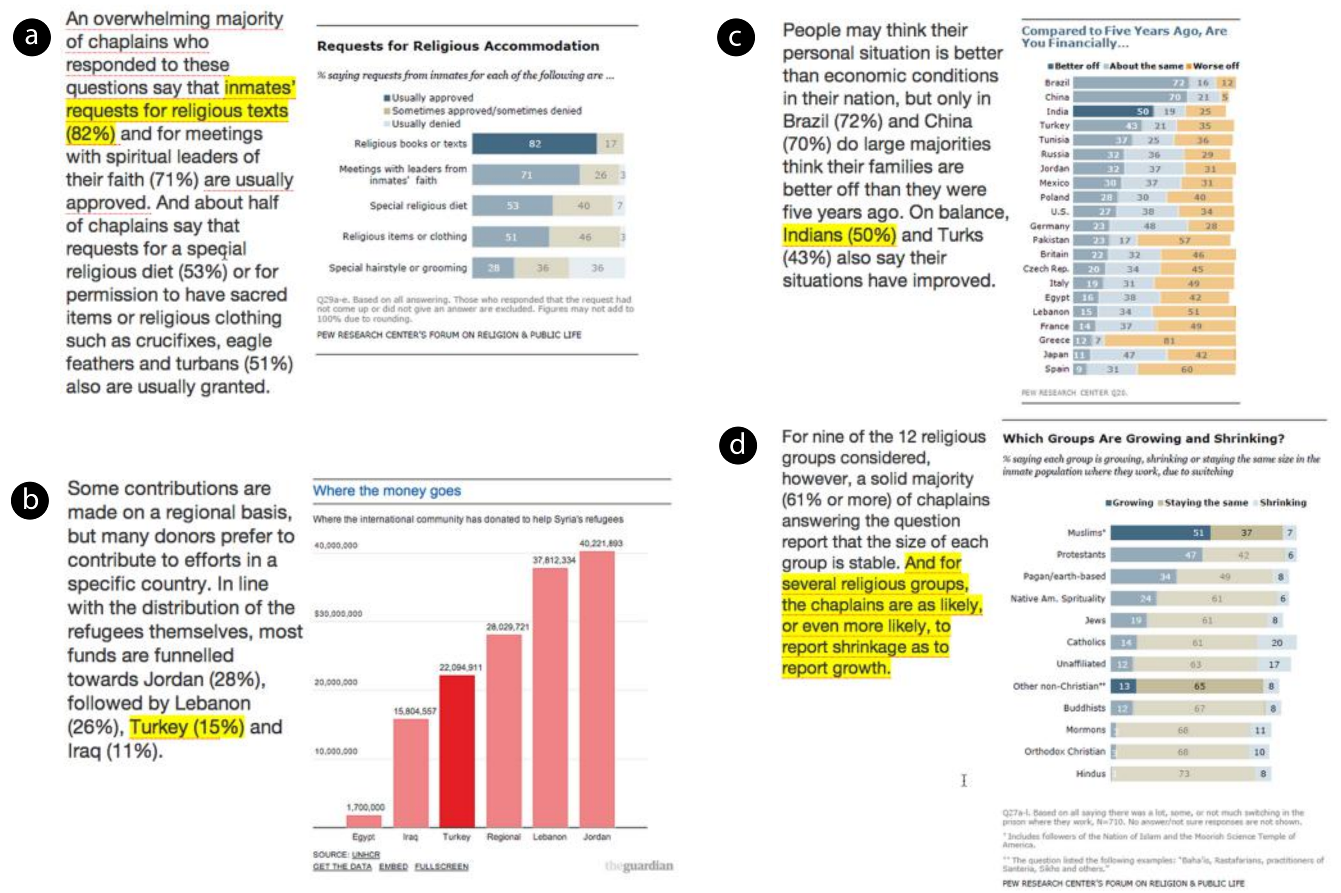
Of the 400 worker generated reference sets, 220 (55%) passed the accompanying gold check. Figure 7a shows the phrase-level distances between each worker's reference set and the gold standard for each of the 40 paragraph-chart conditions. Each column represents one condition and lower points are better because they are closer to the gold standard. The light blue dots mark workers that failed the gold check, while the darker blue circles mark those who passed it. Red circles represent the distances between our final set of unified references after clustering and merging, and the gold standard.

Figure 7a also shows that there is a relatively large spread in the distances between the worker reference sets and the gold reference sets in every condition. Even when we reject the worker references sets that failed the gold check the spread is large. Using our clustering and merging techniques produces references that are closer to the ground truth than the average of the workers (with or without the gold check). However, in most cases the best workers can generate reference sets that are closer to the gold standard (i.e., lower in the chart) than our unified reference sets.

Figure 7b aggregates these distances across all 40 conditions. It shows that the average distance between all of the worker generated references and the gold standard is 0.54. Limiting to just the workers who pass the gold check the average distance reduces to 0.47 and when we apply our our clustering and merging algorithms the average distance reduces still further to 0.39. Our clustering and merging improves the reference sets by 27% compared to all workers and 18% compared to the workers who passed the gold check. We also compute the average precision and recall separately for the text and data for the clustered and merged references. Across the 40 conditions we obtain an average text precision of 0.74, average text recall of 0.69, average data precision of 0.77 and average data recall of 0.83.

Sentence-level references

Our gold standard references are designed to be *minimal* in the sense that they contain the smallest set of text phrases and data tuples that uniquely correspond to one another. However for some applications (e.g., see next section on interactive document viewer) it may be sufficient to recover the correspondence between each sentence in the paragraph and the


a

An overwhelming majority of chaplains who responded to these questions say that inmates' requests for religious texts (82%) and for meetings with spiritual leaders of their faith (71%) are usually approved. And about half of chaplains say that requests for a special religious diet (53%) or for permission to have sacred items or religious clothing such as crucifixes, eagle feathers and turbans (51%) also are usually granted.

b

Some contributions are made on a regional basis, but many donors prefer to contribute to efforts in a specific country. In line with the distribution of the refugees themselves, most funds are funnelled towards Jordan (28%), followed by Lebanon (26%), Turkey (15%) and Iraq (11%).

c

People may think their personal situation is better than economic conditions in their nation, but only in Brazil (72%) and China (70%) do large majorities think their families are better off than they were five years ago. On balance, Indians (50%) and Turks (43%) also say their situations have improved.

d

For nine of the 12 religious groups considered, however, a solid majority (61% or more) of chaplains answering the question report that the size of each group is stable. And for several religious groups, the chaplains are as likely, or even more likely, to report shrinkage as to report growth.

Requests for Religious Accommodation
% saying requests from inmates for each of the following are ...
Usually approved
Sometimes approved/sometimes denied
Usually denied
Religious books or texts 82 17
Meetings with leaders from inmates' faith 71 26 3
Special religious diet 53 40 7
Religious items or clothing 51 46 3
Special hairstyle or grooming 28 36 36
PEW RESEARCH CENTER'S FORUM ON RELIGION & PUBLIC LIFE
Where the money goes
Where the international community has donated to help Syria's refugees
40,000,000
30,000,000
20,000,000
10,000,000
1,700,000
15,804,557
22,094,911
28,029,721
37,812,334
40,221,893
Egypt
Iraq
Turkey
Regional
Lebanon
Jordan
SOURCE: UNHCR
GET THE DATA EMBED FULLSCREEN
theguardian
Compared to Five Years Ago, Are You Financially...
Better off
About the same
Worse off
Brazil 72 16 12
China 70 21 5
India 50 19 25
Turkey 43 21 35
Tunisia 37 25 36
Russia 32 36 29
Jordan 32 37 31
Mexico 30 37 31
Poland 28 30 40
U.S. 27 38 34
Germany 23 48 28
Pakistan 23 17 57
Britain 22 32 46
Czech Rep. 20 34 45
Italy 19 31 49
Egypt 16 38 42
Lebanon 15 34 51
France 14 37 49
Greece 12 7 81
Japan 11 47 42
Spain 9 31 60
PEW RESEARCH CENTER Q25.
Which Groups Are Growing and Shrinking?
% saying each group is growing, shrinking or staying the same size in the inmate population where they work, due to switching
Growing
Staying the same
Shrinking
Muslims* 51 37 7
Protestants 47 42 6
Pagan/earth-based 34 49 8
Native Am. Spirituality 24 61 6
Jews 19 61 8
Catholics 14 61 20
Unaffiliated 12 63 17
Other non-Christian** 13 65 8
Buddhists 12 67 8
Mormons 68 11
Orthodox Christian 68 10
Hindus 73 8
PEW RESEARCH CENTER'S FORUM ON RELIGION & PUBLIC LIFE

Figure 8. Our interactive document viewing application lets users select text (yellow background) and it highlights the corresponding visual marks in the chart (fully saturated bars). The application also places red underlines beneath related phrases. In all four cases the viewer is highlighting marks based on our crowdsourced references after clustering and merging. Examples (a), (b) and (c) show paragraph-chart pairs in which our pipeline creates high quality references. Example (d) shows a paragraph-chart pair for which our pipeline did not extract the correct reference.

data tuples. Such references are more robust than phrase-level references since they do not penalize extraneous or missing words within a sentence.

We compute a sentence-level distance to our gold standard references as follows. We first split the paragraph text into sentences using the Punkt sentence tokenizer in NLTK [4]. For each reference in a reference set (either gold or worker generated), we replace the phrases with the sentences that contain them. We then form a vector with an entry for each possible sentence-tuple pair and we set an entry to true if the sentence and tuple refer to each other (i.e., are contained in the same reference) and set it to false otherwise. Finally, we compute F_1 scores between the sentence-tuple vectors corresponding to the gold and worker gnerated reference sets, and then convert the score into a distance by computing $1 - F_1$.

Figure 7c shows the sentence-level distances between each worker's reference set and the gold standard for each condition. The pipeline is more successful at extracting sentence-level references than phrase-level references by these measures, although we see the same variability across conditions. Figure 7d shows average sentence-level distances to the gold. We find that the average distance for worker references to the gold is 0.35, while our clustering and merging pipeline creates reference sets with an average distance of 0.22, an improvement of 37%.

APPLICATION: INTERACTIVE DOCUMENT VIEWER

Reading a document and correctly identifying the references between the text and the chart can be difficult (Figure 1). We have developed a proof-of-concept interactive document viewing application that uses the references generated by our crowdsourced extraction pipeline to explicitly highlight these correspondences. As shown in Figures 1 and 8 our implementation presents a document as a paragraph of text and a chart. The user can click and drag to select text and our application highlights the corresponding data-encoding marks in the chart by fading out the surrounding marks.

More specifically we identify all extracted references whose text phrases are fully contained within the selected text. We then look up the data tuples for all such references and highlight the corresponding visual marks in the chart. If we find that there are no references with text phrases that are fully contained within the selected text we use a more lenient highlighting strategy. We identify all of the references whose text contains at least one word of the selected text and then highlight the visual marks for all of the data tuples in the resulting set of references. We implement the mark highlighting as a graphical overlay on the chart bitmap using the approach of Kong et al. [19]. We also draw red underlines beneath all of the text phrases in the references that we highlight to further help readers connect the text with the chart.

While Figures 1 and 8 show our application working with our clustered and merged reference sets, the application can also be used to explore the raw worker generated reference sets, as well as the gold standard reference sets. The application also optionally expands users' highlighted phrases to full sentences to display sentence-level references. Reference sets and the interactive application are available at `http://vis.berkeley.edu/papers/textref/supplemental`.

DISCUSSION

The large spread of distances in Figure 7 between the raw worker generated references and the gold standard suggests that extracting references for some paragraph-chart pairs can be easier than for others. We noticed that workers were most successful when short text phrases in a paragraph corresponded with a single data tuple (e.g., one visual mark).

Consider the examples in Figure 8. In example (a), the text phrase "inmates' request for religious texts (82%)" corresponds to a single blue bar that is labeled with the words *religious books or texts* and the number *82*. Similarly in examples (b) and (c) text phrases "Turkey (15%)" and "Indians (50%)" correspond to bars labeled *Turkey* and *India* respectively. In these cases the simple text phrases and the labels in the chart make it relatively easy to correctly extract the references. Indeed, these three examples correspond to paragraph-chart conditions 9, 76 and 28 respectively. Looking them up in Figure 7 (left) we see that workers who passed the gold check extracted relatively high-quality references that were close to the gold standard (dark blue circles are close to zero).

Example (d) is much more complicated. The text phrase "And for several religious groups, the chaplains are as likely, or even more likely, to report shrinkage as to report growth." refers to the set of bar segments for religions in which shrinkage is larger than or equal to growth. In this case the set of visual marks corresponding to this text include the *growing* and *shrinking* bar segments for *Catholics, Unaffiliated, Mormons, Orthodox Christian* and *Hindus*. Moreover the minimal text phrases for this reference only includes the words "several", "as likely, or even more likely", "shrinkage" and "growth". All the other words in the full sentence are extraneous. This challenging example corresponds to condition 6 and as shown in Figure 7 (left) the workers who passed the gold check for this condition produced relatively poor-quality references.

Figure 1 also shows a complicated reference for condition 30 in which the text refers to 13 bar segments in the chart. While many of the workers generated mediocre quality references for this condition, our clustering and merging algorithms were able to combine the best information from the workers and extract a unified set of references that are much better than any of the individual worker generated reference sets. Nevertheless, an open direction for future work is to develop techniques to help crowd workers correctly extract references in such complicated cases.

CONCLUSION AND FUTURE WORK

We have introduced the problem of extracting the references between the text and charts in a document and a pipeline that combines crowdsourcing with clustering and merging algorithms to extract such references. We have also shown how such references can be used to facilitate reading with a proof-of-concept application for interactive document viewing.

We believe that extracting an accurate set of references for a document opens the door to variety of applications. Our interactive document viewer lets readers select text and see the corresponding visual marks. One extension is to let readers select marks in the chart and see where they are referenced in the text. It may be possible to improve document accessibility based on the references. For example a screen reader for charts could pull in the relevant surrounding text as it described the chart. The references could be used to aid document authors; as an author writes text and designs figures such an aid might warn authors if references between the text and charts are missing (e.g., the author wrote about a bar that doesn't appear in the chart).

We also believe that the crowdsourcing pipeline could be further improved to help workers accurately extract complex reference with text phrases that are non-contiguous and refer to multiple data tuples. One approach may be to include a separate crowdsourced stage in which an independent set of workers votes on the correctness of the references produced by an initial set of workers.

It may also be possible to automatically extract references using a combination of natural language processing, computer vision and machine learning techniques. However, our results suggest that extracting high-quality, minimal references between text and charts is challenging even for humans and doing it automatically is likely to require sophisticated processing techniques. Thus, it may be most fruitful to combine these algorithmic techniques with human intervention.

Acknowledgements: We thank Wesley Willett for his assistance with the gold standard. This work was supported by NSF grants IIS-1016920, IIS-1252819, and an NSERC PGS D scholarship.

REFERENCES

1. Ahmad, S., Battle, A., Malkani, Z., and Kamvar, S. The jabberwocky programming environment for structured social computing. In *Proc. of UIST* (2011), 53–64.
2. 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 *Proc. of UIST* (2010), 313–322.
3. Bernstein, M. S., Teevan, J., Dumais, S., Liebling, D., and Horvitz, E. Direct answers for search queries in the long tail. In *Proc. of SIGCHI* (2012), 237–246.
4. Bird, S. NLTK: The natural language toolkit. In *Proc. of COLING/ACL Interactive Presentations* (2006), 69–72.
5. Cairo, A. *The Functional Art: An introduction to information graphics and visualization*. New Riders, 2012.

6. Callison-Burch, C., and Dredze, M. Creating speech and language data with amazon's mechanical turk. In *Proc. of NAACL HLT Workshop on Creating Speech and Language with Amazon's Mechanical Turk* (2010), 1–12.

7. Card, S. K., Mackinlay, J. D., and Shneiderman, B., Eds. *Readings in information visualization: Using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

8. Chilton, L. B., Little, G., Edge, D., Weld, D. S., and Landay, J. A. Cascade: Crowdsourcing taxonomy creation. In *Proc. of SIGCHI* (2013), 1999–2008.

9. Cleveland, W. S. Elements of graphing data.

10. Economist Graphic Detail. `http://www.economist.com/blogs/graphicdetail`. Retrieved Sep 17, 2013.

11. Guardian DataBlog. `http://www.theguardian.com/datablog`. Retrieved Sep 17, 2013.

12. Heer, J., and Bostock, M. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proc. of SIGCHI* (2010), 203–212.

13. Hsueh, P.-Y., Melville, P., and Sindhwani, V. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proc. of NAACL HLT Workshop on Active Learning for Natural Language Processing* (2009), 27–35.

14. Hullman, J., Adar, E., and Shah, P. The impact of social information on visual judgments. In *Proc. of SIGCHI* (2011), 1461–1470.

15. Hullman, J., and Diakopoulos, N. Visualization rhetoric: Framing effects in narrative visualization. *IEEE TVCG 17*, 12 (2011), 2231–2240.

16. Hullman, J., Diakopoulos, N., and Adar, E. Contextifier: Automatic generation of annotated stock visualizations. In *Proc. of SIGCHI* (2013), 2707–2716.

17. Kandogan, E. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In *IEEE VAST* (2012), 73–82.

18. Kittur, A., Smus, B., Khamkar, S., and Kraut, R. E. Crowdforge: Crowdsourcing complex work. In *Proc. of UIST* (2011), 43–52.

19. Kong, N., and Agrawala, M. Graphical overlays: Using layered elements to aid chart reading. *IEEE TVCG 18*, 12 (2012), 2631–2638.

20. Kong, N., Heer, J., and Agrawala, M. Perceptual guidelines for creating rectangular treemaps. *IEEE TVCG 16*, 6 (2010), 990–998.

21. Kulkarni, A., Can, M., and Hartmann, B. Collaboratively crowdsourcing workflows with turkomatic. In *Proc. CSCW* (2012), 1003–1012.

22. Le, J., Edmonds, A., Hester, V., and Biewald, L. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation* (2010), 21–26.

23. Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. TurKit: Human computation algorithms on mechanical turk. In *Proc. of UIST* (2010), 57–66.

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

25. Mayfield, J., Lawrie, D., McNamee, P., and Oard, D. Building a cross-language entity linking collection in twenty-one languages. In *Multilingual and Multimodal Information Access Evaluation*. Springer, 2011, 3–13.

26. Nakov, P. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2008, 103–117.

27. Oleson, D., Sorokin, A., Laughlin, G., Hester, V., Le, J., and Biewald, L. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Proc. of HComp* (2011).

28. Pew Research. `http://www.pewresearch.org/`. Retrieved Sep 17, 2013.

29. Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., and Heer, J. Revision: Automated classification, analysis and redesign of chart images. In *Proc. of UIST* (2011), 393–402.

30. Segel, E., and Heer, J. Narrative visualization: Telling stories with data. *IEEE TVCG 16*, 6 (2010), 1139–1148.

31. Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP* (2008), 254–263.

32. Substance Abuse and Mental Health Services Administration. `http://www.samhsa.gov/`. Retrieved Sep 17, 2013.

33. Willett, W., Heer, J., and Agrawala, M. Strategies for crowdsourcing social data analysis. In *Proc. of SIGCHI* (2012), 227–236.

APPENDIX A: GOLD REFERENCE GUIDELINES

Two experts converged on the following guidelines:

- Each reference should contain as little text as necessary to explicitly specify a relation to one or more tuples.
- Each reference should have a different phrase set.
- Ignore text that refers to the chart as a whole (e.g., the phrase "religious extremism" in a chart where every tuple is related to religious extremism).
- In the case of an ambiguous phrase that refers to an ambiguous subset of a chart (e.g., "the rich" or "the poor"), make a best effort guess.