CrossMark

Crowdsourcing chart digitizer: task design and quality control for making legacy open data machine-readable

Satoshi Oyama[1] · Yukino Baba[2] · Ikki Ohmukai[3] · Hiroaki Dokoshi[1] · Hisashi Kashima[2]

Abstract Despite recent open data initiatives in many countries, a significant percentage of the data provided is in non-machine-readable formats like image format rather than in a machine-readable electronic format, thereby restricting their usability. Various types of software for digitizing data chart images have been developed. However, such software is designed for manual use and thus requires human intervention, making it unsuitable for automatically extracting data from a large number of chart images. This paper describes the first unified framework for converting legacy open data in chart images into a machine-readable and reusable format by using crowdsourcing. Crowd workers are asked not only to extract data from an image of a chart but also to reproduce the chart objects in a spreadsheet. The properties of the reproduced chart objects give their data structures, including series names and values, which are useful for automatic processing of data by computer. Since results produced by crowdsourcing inherently contain errors, a quality control mechanism was developed that improves accuracy by aggregating tables created by different workers for the same chart image and by utilizing the data structures obtained from the reproduced chart objects. Experimental results demonstrated that the proposed framework and mechanism are effective. The proposed framework is not intended to compete with chart digitizing software, and workers can use it if they feel it is useful for extracting data from charts. Experiments in which workers were encouraged to use such software showed that even if workers used it, the extracted data still contained errors. This indicates that quality control is necessary even if workers use software to extract data from chart images.

Keywords Crowdsourcing · Open data · Statistical chart · Data extraction · Spreadsheet

This paper is an extended version of the DSAA 2015 long presentation paper "From one star to three stars: upgrading legacy open data using crowdsourcing" [1].

✉ Satoshi Oyama
   oyama@ist.hokudai.ac.jp

   Yukino Baba
   baba@i.kyoto-u.ac.jp

   Ikki Ohmukai
   i2k@nii.ac.jp

   Hiroaki Dokoshi
   dokoshi@complex.ist.hokudai.ac.jp

   Hisashi Kashima
   kashima@i.kyoto-u.ac.jp

[1] Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060–0814, Japan

[2] Graduate School of Informatics, Kyoto University, 36-1 Yoshida-Honmachi, Sakyo-ku, Kyoto 606–8501, Japan

[3] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

1 Introduction

The most prominent of the recent open data initiatives to publish various kinds of data in electronic format is ones for statistical data gathered by governmental agencies [2]. Publishing such data is expected to improve government transparency, facilitate citizen participation, and create new business opportunities. These recent initiatives have led to the creation of data catalog sites by many countries, including the USA[1], the UK[2], and Japan[3], that provide data under an open reuse license.

---

[1] http://catalog.data.gov/.

[2] http://data.gov.uk/.

[3] http://www.data.go.jp/.

Tim Berners-Lee, the creator of the Web, developed a star rating scheme to encourage the publishing of data[4]:

- The data are available on the Web (in whatever format) with an open license.
- The data are available in a machine-readable structured format (e.g., Microsoft Excel) instead of an image format.
- The first two plus the data are in a non-proprietary format (e.g., CSV).
- The first three plus open standards from the World Wide Web Consortium (W3C), the Resource Description Framework (RDF) with the SPARQL Protocol and RDF Query Language (SPARQL) are used to identify things.
- The first four plus the data are linked to other people's data to provide context.

However, a significant percentage of published statistical data was published as charts or graphs in image or PDF files. For example, of the 10,410 datasets provided by the Japanese government data site, 5452 are provided as PDF files. In the US data catalog site, 4838 of the 104,793 datasets are provided as PDF files. Such datasets earn only one star in Berners-Lee's rating scheme and are not readily reusable because extracting data from figures and tables in PDF files is not easy even if they are provided with open licenses. One of the major reasons for such hasty data publishing was limited budgets and human resources in governmental agencies. They cannot afford to convert such data into machine-readable formats by themselves. The percentage of data published in machine-readable formats, such as CSV and RDF, will increase, but a certain amount of data will continue to be published in PDF or image files for a while. Furthermore, legacy data are generally published in such formats.

There have been certain demands for extracting values from statistical charts among the scientific community, typically for reusing data published in old papers. To meet such demands, various types of chart digitizing software such as WebPlotDigitizer[5] and DataThief[6] have been developed. However, such software is designed for manual use and thus requires human intervention, such as in calibrating the chart axes, making it unsuitable for automatically extracting data from a large number of data charts.

Figure 1 shows examples of charts used in the 2013 White Paper on Tourism[7], which was published by the Japan Tourism Agency. Some charts are very complicated or in nonstandard formats. For example, in the line chart (b), lines

representing different data series cross each other. The pie chart (c) is represented as a cylinder in a 3D space rather than a simple circle. In chart (f), both bars and lines are used to represent the data. Such variety and complexity make it difficult to automatically extract data using chart digitizing software.

Human computation [3] has been attracting attention as a way to solve problems that are difficult to solve solely with a computer but are solvable with some human help. The increasing size of the workforce available online provided by crowdsourcing services such as Amazon Mechanical Turk[8] has been pushing widespread application of human computation to data processing tasks in various domains. This will help overcome the bottleneck in promoting open data due to limited human resources. Open data are especially suitable for being processed by human computation because the data confidentiality/privacy issue, which is sometimes a barrier to crowdsourced data processing, does not have to be considered.

Data charts are designed to help people better understand data, and people are better at understanding them than computers. We have thus taken a human computation approach to the *datafication* of legacy data: use crowdsourcing to extract structured data from charts in legacy file formats such as image and PDF files. Doing this will improve the ranking of such data from one star in Berners-Lee's scheme to two or three stars. To the best of our knowledge, this paper presents the first unified framework for converting legacy open data in chart images into a machine-readable, reusable format by using crowdsourcing. Also presented is a quality control mechanism that improves the accuracy of extracted tables by aggregating tables created by different workers for the same chart image and by utilizing the data structures obtained from the reproduced chart objects. Testing showed that the proposed framework and mechanism are effective.

The goal of this paper is to demonstrate the feasibility of our approach in terms of the four components of crowdsourcing: the requester, the crowd, the task, and the platform [4]. Note that our crowdsourcing framework was not developed to compete with chart digitizing software. In fact, such software is not excluded from our framework, and workers can use it if they feel it is useful for extracting data from charts. Since current chart digitizing software needs human intervention and thus is susceptible to human error, quality control is needed to obtain high-quality results. In that sense, our proposed framework is independent of using/not using chart digitizing software. We include experiments in which workers were encouraged to use chart digitizing software as well.

The remainder of the paper is organized as follows. Section 2 discusses related work on open data and crowdsourcing. Section 3 describes our framework for digitizing
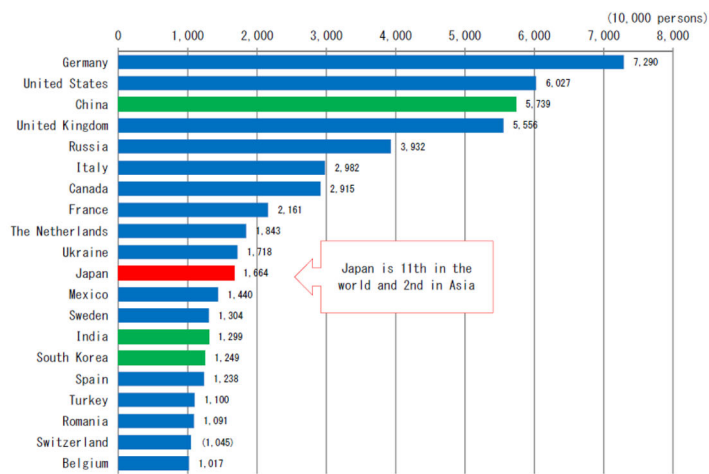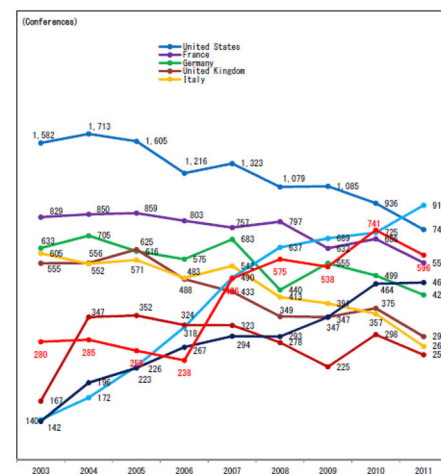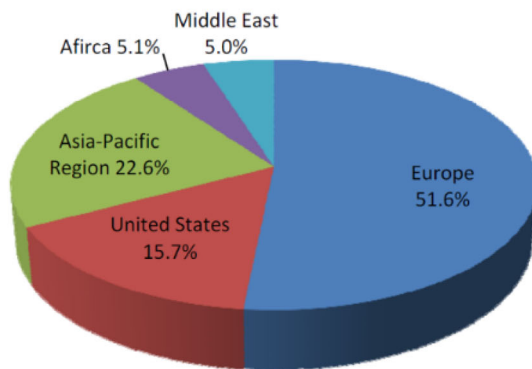
---

4 http://www.w3.org/DesignIssues/LinkedData.html.

5 http://arohatgi.info/WebPlotDigitizer/.

6 http://datathief.org/.

7 http://www.mlit.go.jp/statistics/file000008.html.

8 http://aws.amazon.com/mturk/.
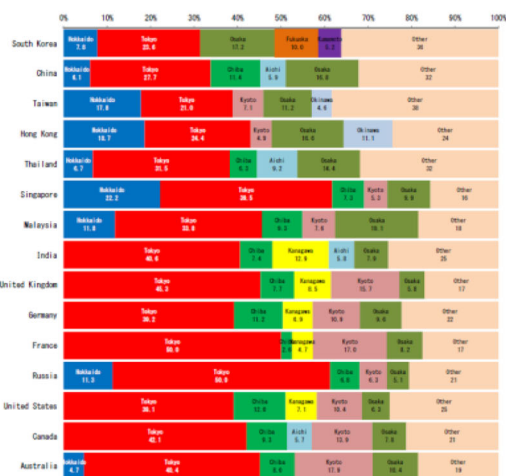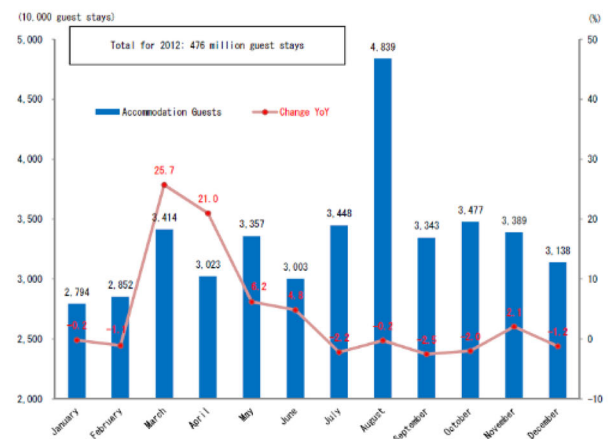
(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1 Examples of charts used in 2013 White Paper on Tourism published by the Japan Tourism Agency **a** bar chart, **b** line chart, **c** pie chart, **d** stacked bar chart, **e** 100 % stacked bar chart, **f** bar+line chart

chart images using crowdsourcing, and Sect. 4 details the quality control mechanism used in our framework. Section 5 presents experiments using real-case data to demonstrate the feasibility and effectiveness of our framework, and Sect. 6 shows experimental results when we encouraged workers to use chart digitizing software within our framework. Finally, Sect. 7 summarizes the key points and describes future research directions.

2 Related work

Studies to promote open data in accordance with the road map put forth by Berners-Lee have been conducted by researchers investigating the Semantic Web. However, their research interests have focused mainly on pulling three-star data up to the fourth or fifth levels and building services on top of them—few have focused on dealing with one-star data.

Han et al. [5] developed an open-source tool for converting spreadsheet data into RDF format. Users define the relationships between the columns of a spreadsheet table in a *map graph* by using a graphical interface. A Web service then takes the spreadsheet or CSV file and the map graph and provides an RDF file as output. Mulwad et al. [6] presented techniques for automatically inferring the semantics of column headers and cell values and the relationships between columns. Their techniques are based on graphical models and probabilistic reasoning augmented with background knowledge from the Linked Open Data cloud (Web of Linked Data [7]).

The RDF Data Cube Vocabulary[9] is a W3C recommendation for publishing and sharing statistical data on the Web. In the Data Cube model, data observations (values in table cells) are characterized by dimensions, attributes, and measures. Meroño-Peñuela et al. [8] converted historical census data into RDF format by using the Data Cube Vocabulary and a semiautomatic process: First, an expert manually annotated tables in Microsoft Excel workbooks, and then software called TabLinker[10] was used to convert them into RDF data cubes.

Government data made public with open licenses are called *open government data (OGD)* and are considered important for enhancing the transparency of governance and improving public services by promoting participatory decision making. Shadbolt et al. [2] described a project for integrating CSV data and spreadsheet data published on the UK data catalog site in the Web of Linked Data. Kalampokis et al. [9] asserted that the real value of OGD comes from performing data analytics on top of combined datasets from different sources. As a test case, they used various published

datasets including the unemployment rate dataset published in spreadsheet form on data.gov.uk and datasets regarding the outcome of UK general elections published in spreadsheet form on the Guardian newspaper's Web site under an open license. They converted the datasets into RDF data cubes and performed data analytics using the combined datasets. They showed that there was a correlation between the probability of one of the two main political parties winning a particular constituency and the unemployment rate for that constituency.

Crowdsourcing is a means for asking many workers to perform tasks via the Internet. There are various types of crowdsourcing, which can be classified in terms of workers, requesters, tasks, and platforms [4]. Crowd workers can be paid via a crowdsourcing market such as Amazon Mechanical Turk or be volunteers. In the latter case, games with a purpose [10] are usually used. Human computation is a computing paradigm used to solve problems that are difficult to solve solely by computer but are relatively easy to solve if human assistance is provided. It is generally achieved by asking people to complete small tasks, called *microtasks* or *human intelligence tasks*, by crowdsourcing. Human computation approaches have been successfully applied to various domains including computer vision and natural language processing, for which current artificial intelligence is still no match for human intelligence. The importance of explicit or implicit involvement of human intelligence is also emphasized in metasynthetic engineering of complex systems [11], in which human intelligence and machine intelligence are integrated into problem-solving systems. Human intelligence is expected to provide qualitative reasoning capability and complement quantitative machine intelligence.

One of the most successful examples of human computation is reCAPTCHA [12], which is a system both for verifying that an on-line user is actually human and for deciphering words unrecognized by optical character recognition (OCR) software used in, for example, a book digitization project. Two image files are presented to the user, one containing a word known by the computer and used for verification, and one containing an unrecognized word. The user must type the characters shown in both files. If the known word is typed correctly, the user is recognized as human. If enough users interpret the unknown word as a certain word, that word is considered valid. This process can be considered to be digitizing characters in images using human computation.

A British newspaper *The Guardian* conducted experiments to analyze the receipts of the Members of Parliament using crowdsourcing.[11] Photocopies of handwritten receipts

---

were provided in image files, and users were asked to review each of them and type the content, label them in terms of level of interest, and write some comments. This can be considered as an example of datafication of open data provided in image files. In contrast, our work focuses on digitizing statistical charts in image files.

AskSheet [13] is a system that uses crowdsourcing to make spreadsheet tables. It does not ask workers to make an entire table but instead asks them to gather the information needed to fill in the blank cells in a spreadsheet. As with our approach, it uses a quality control mechanism to compensate for erroneous inputs by workers, but the mechanism is applied to individual values while ours is applied to the entire table. Fan et al. [14] proposed a framework for utilizing crowd workers to aggregate tables. They prepared a concept catalog and asked the workers to select the concept that best represented the values in each column. The selected concepts were used to match the columns that appeared in different tables but corresponded semantically. They focused only on one-dimensional tables and assumed that there were no incorrect values in the tables. In contrast, we focused on two-dimensional tables and assumed that there were incorrect values. Ermilov et al. [15] proposed a formalization of tabular data as well as its mapping and transformation to RDF, which enable the crowdsourcing of large-scale semantic mapping of tabular data. They mentioned automatic header recognition in CSV files as important future work. In our approach, headers in tables can be easily recognized by referring to the properties of the chart objects in the spreadsheets.

The Semantic Web is an endeavor to make the Web machine-understandable, so both machine and human intelligence are naturally needed to achieve it. Human computation practices have been applied to various subproblems as part of enabling the Semantic Web [16–20].

3 Framework for digitizing chart images using crowdsourcing

In this section we describe our framework for digitizing chart images using crowdsourcing. We present our task design, which makes it easy to extract structured data, and discuss the feasibility of our framework in terms of the requester, the workers, the tasks, and the platform.

3.1 Task design

One of the most important aspects of human computation is designing the task so that intuitive and effective instructions can be given to non-expert workers so that they can work efficiently and accurately. The objective in our study was extraction of data from charts in image format and

conversion of them into a form convenient for computer processing. A possible task design is to simply ask workers to extract graph data and place them in a CSV- or Excel-formatted file; however, the output with this approach does not provide a data structure, such as a distinction between row/column headers and data values, which is inconvenient for later data processing steps like data integration. Therefore, in our method, workers are asked to instead visually reproduce a chart image as a chart object in a spreadsheet using the functions of the spreadsheet software. This enables us to obtain a table linked to a chart object representing the data in the table and obtain the structure of the data, such as row and column headers and data sequences, from the properties of the chart object. It is not a straightforward task to automatically identify row and column headers in a table in a CSV file or a spreadsheet without the chart, but they can be easily obtained using an application programming interface if the chart object is provided with the table. This task design (Fig. 2) is an example of having an implicit purpose (extracting structured data) hidden behind an explicit instruction (reproducing a chart), which is common in human computation such as in the reCAPTCHA system [3].

Additionally, the structure of the data is essential for controlling the quality of digitizing work; it provides an efficient way to aggregate tables made by different workers and enables use of the common practice of asking multiple workers to complete the same task and then aggregating the results. Figure 3 shows our framework for digitizing chart images using crowdsourcing. The inputs are charts in an image file format such as JPEG. Microtasks asking crowd workers to reproduce the images in spreadsheets are generated and posted to a crowdsourcing marketplace. Several workers are assigned to each image. Each worker converts the image into a spreadsheet (in Microsoft Excel format) with an embedded graph. The axis labels, legends, and sequence values are extracted from the submitted file, resulting in pairs of attribute names and values. Finally, the spreadsheets obtained from the different workers are integrated into a single, higher-quality spreadsheet.

While Berners-Lee ranks CSV above Microsoft Excel since the former is an open format while the latter uses a proprietary one, in practice, the distinction is not substantial because recent versions of Excel use the Office Open XML format, and data in this format are readable by other software. Thus, we use Excel spreadsheets as the format in which to save data extracted from chart images.

3.2 Structured data extraction through visualization

During the process of visually reproducing a chart image, a worker has to specify the properties of the chart object in the spreadsheet to reflect the structure of the data repre-

Fig. 2 Task design for extracting data from charts in image format. Workers are asked to visually reproduce a chart image as a chart object in a spreadsheet

Explicit instruction:
reproduce chart
in a spreadsheet

Implicit purpose:
extract data from image
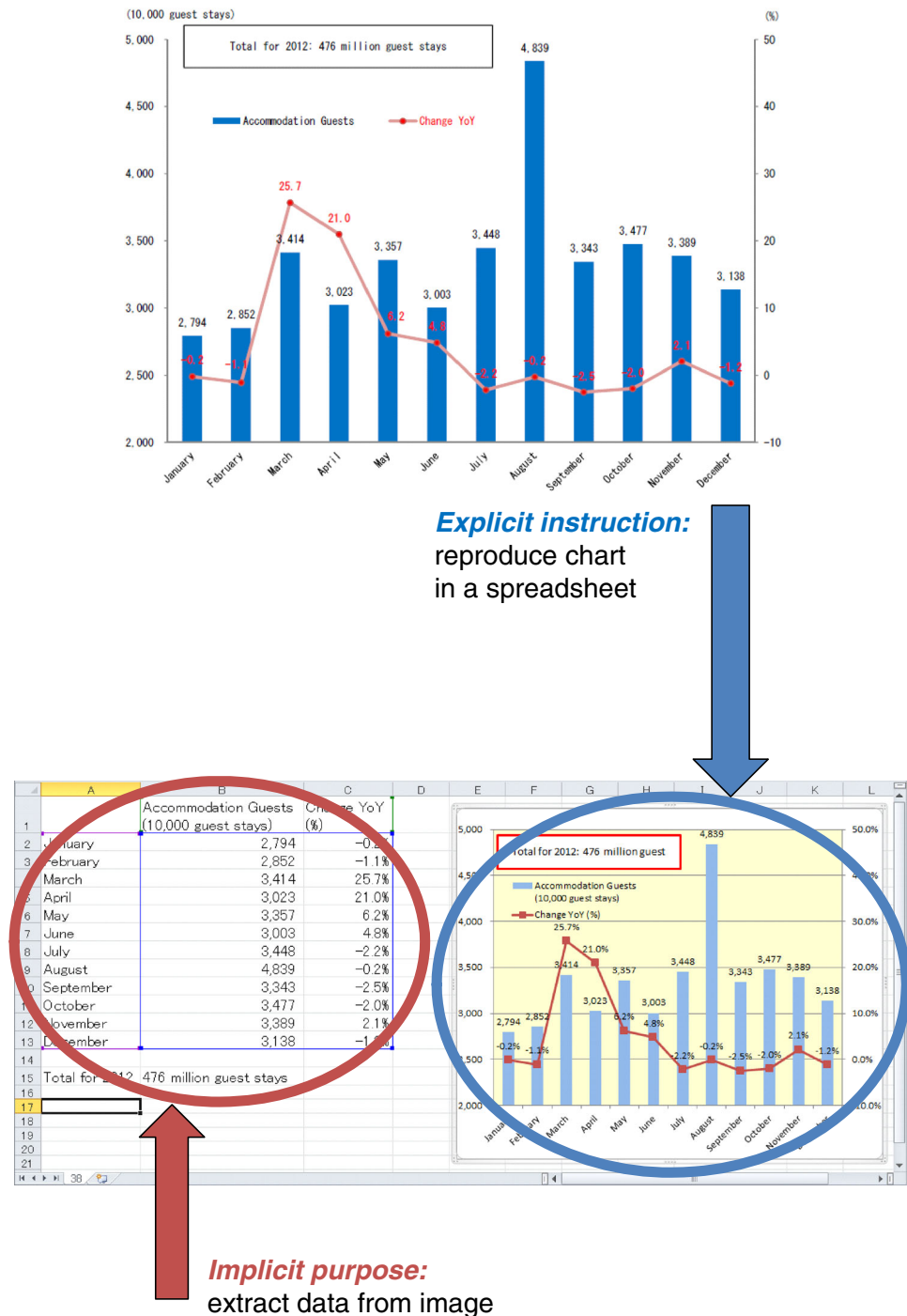
sented in the chart. Such properties can be accessed by using a computer program and an application programming interface. Although there are various kinds of charts including bar charts and line charts, most spreadsheets use a common format for their internal representations; for example, Microsoft Excel uses a three-item format.[12]

– A chart (`Chart`) has several data series (`Series`).
– Each data series (`Series`) has a name (`Name`).
– Each data series (`Series`) has x-axis values (`XValues`) and values (`Values`).

Figure 4 shows the relationships between the structure of a table and the properties of a `Chart` object. Although a two-dimensional table has several possible chart representations, the column labels and row labels correspond to

---

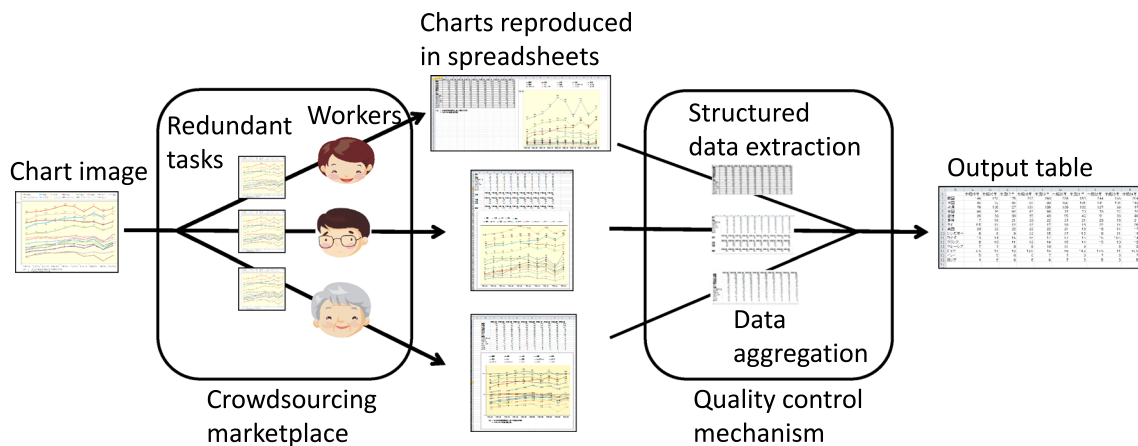[12] http://msdn.microsoft.com/en-us/library/office/ff194068.

Fig. 3 Framework for digitizing chart image using crowdsourcing. Microtasks are generated and posted to a crowdsourcing marketplace. Several workers are assigned to each task. Each worker converts the image into a spreadsheet. The spreadsheets obtained from the different workers are integrated into a single, higher-quality spreadsheet
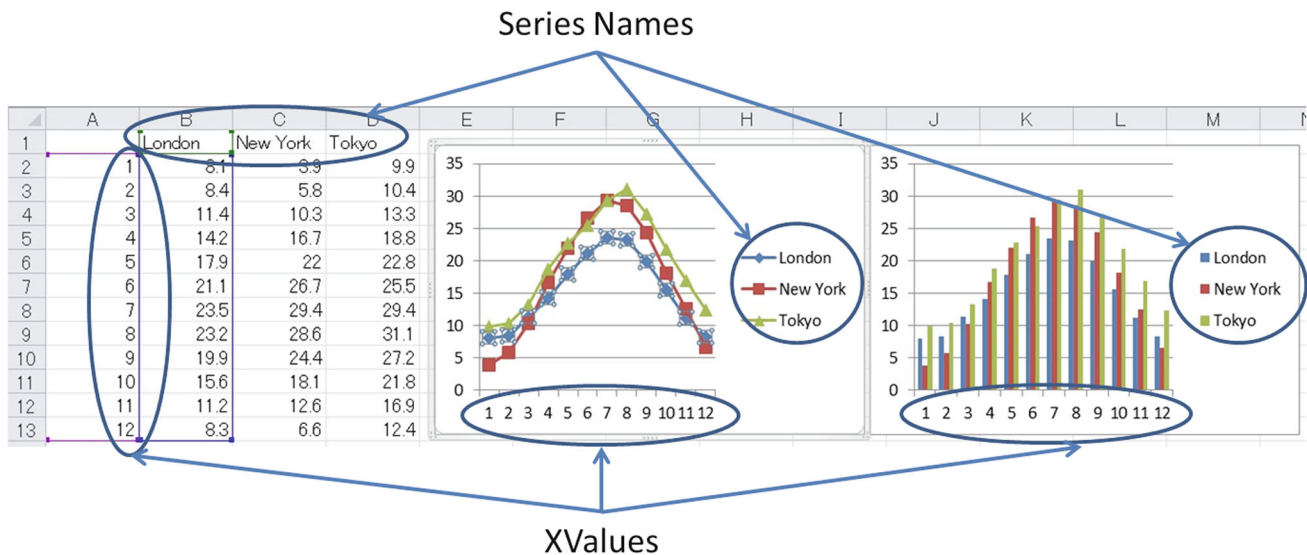

Fig. 4 Relationships between structure of table and properties of `Chart` object. Although a two-dimensional table has several possible chart representations, the column labels and row labels correspond to the labels of the *x*-axis and the legends

the labels of the *x*-axis and the legends; they are extracted as `XValues` and `Name`. The data values are extracted as `Values` in `Series` objects. In tables, the choice of columns and rows is arbitrary; for example, with Figs. 5 and 6, the data categories can correspond to the rows and the months can correspond to the columns and vice versa. In either case, `Name` corresponds to categories, and the `XValues` property corresponds to months; this is essential information for integrating multiple tables since the choice of rows and columns does not need to be made. Moreover, such information is also beneficial when converting tables into RDF format using the RDF Data Cube Vocabulary, which is the next step toward 5-star open data.

3.3 Feasibility of our crowdsourcing framework

According to Hosseini et al. [4], crowdsourcing can be classified on the basis of its four components: the crowdsourcer, the crowd, the crowdsourced task, and the crowdsourcing platform. In our framework, the crowdsourcer (requester) is a governmental agency that owns legacy data. Such an agency has a clear motivation for using crowdsourcing since it does not have enough human resources to convert legacy data into a machine-readable format, and crowdsourcing can reduce the monetary cost of converting the data.

The crowd in our case consists of people who can use spreadsheet software such as Microsoft Excel. The key to the success of our crowdsourcing approach is the availability of
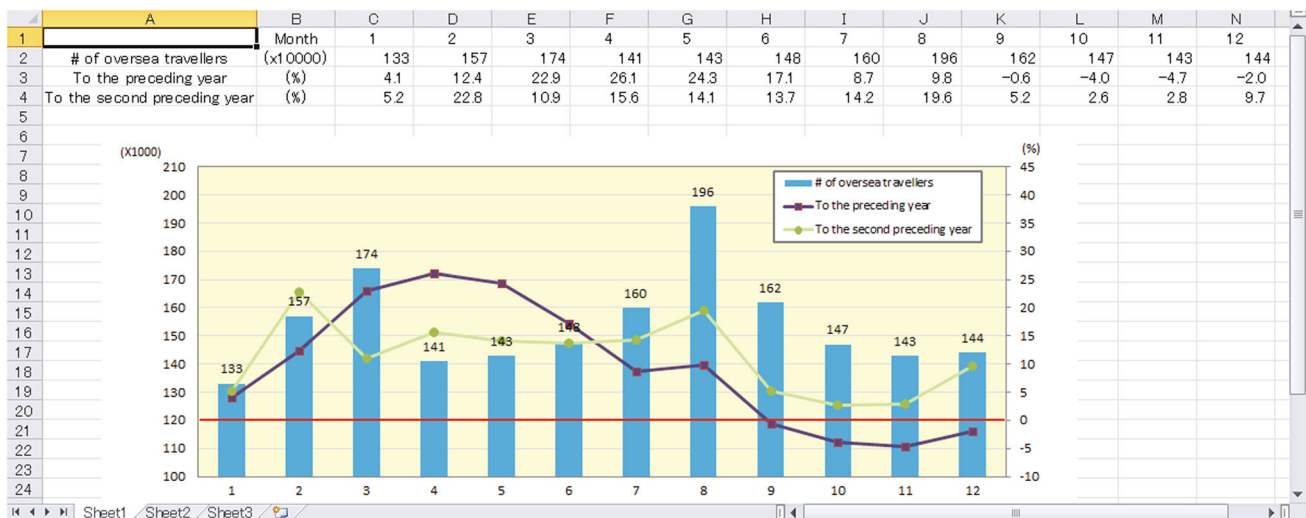
	Month	1	2	3	4	5	6	7	8	9	10	11	12
# of oversea travellers	(x10000)	133	157	174	141	143	148	160	196	162	147	143	144
To the preceding year	(%)	4.1	12.4	22.9	26.1	24.3	17.1	8.7	9.8	−0.6	−4.0	−4.7	−2.0
To the second preceding year	(%)	5.2	22.8	10.9	15.6	14.1	13.7	14.2	19.6	5.2	2.6	2.8	9.7

Fig. 5 Example of table in which data categories correspond to rows and months correspond to columns

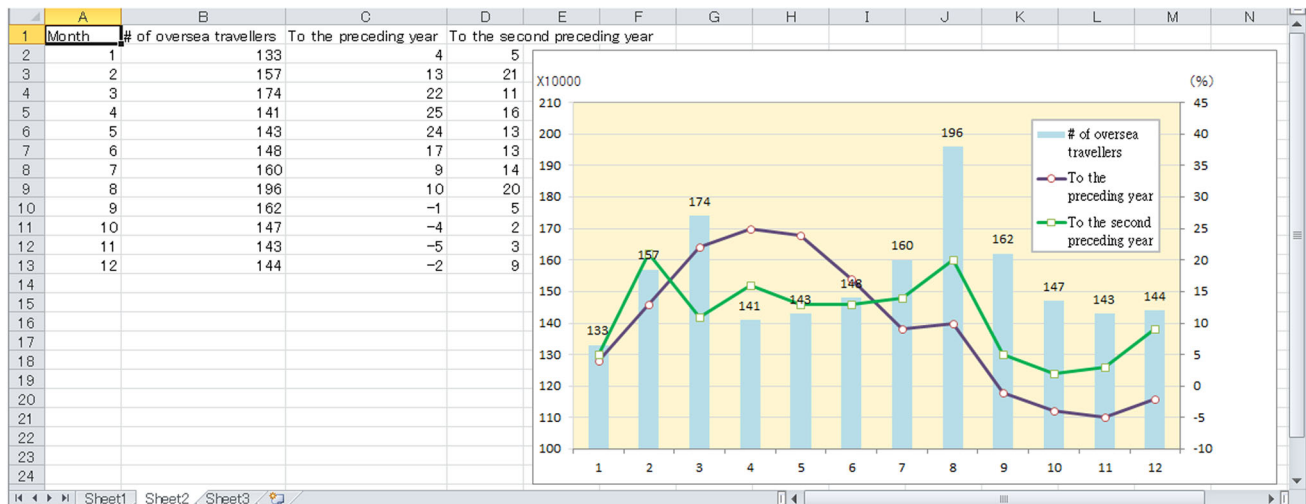| Month	# of oversea travellers	To the preceding year	To the second preceding year
1	133	4	5
2	157	13	21
3	174	22	11
4	141	25	16
5	143	24	13
6	148	17	13
7	160	9	14
8	196	10	20
9	162	−1	5
10	147	−4	2
11	143	−5	3
12	144	−2	9

Fig. 6 Example of table in which data categories correspond to columns and months correspond to rows

such users. Fortunately, Excel is one of the most commonly used business software products worldwide. While we do not have the precise number of Excel users, as of November 2014 more than 1.2 billion people were using one or more of the Office products, which include Excel.[13] Among the 109,344 workers registered with Lancers[14], a crowdsourcing marketplace in Japan, 17,917 have "Excel" in their profiles. The task of extracting data from a chart image is not a simple microtask but a complex task that requires certain software skills. In Sect. 5, we demonstrate that we were able to gather workers with the skills needed to complete the task.

In our case study, we used Lancers but we can use any crowdsourcing platform that has basic functions such as task assignment and reward payment. The quality control mechanism using data aggregation is performed only on the requester's computer using Visual Basic .NET (VB.NET) software. Thus, our framework can be easily implemented by most governmental agencies.

4 Quality control mechanism

Human-created tables may contain errors such as typos and missing values. This is especially true for crowdsourcing as the workers are usually anonymous and not well trained. The requester thus has to take into account errors and introduce a quality control mechanism that makes the results error-tolerant. A common approach to quality control in crowdsourcing is introducing redundancy by asking multiple workers to do the same task. For example, in classification

---

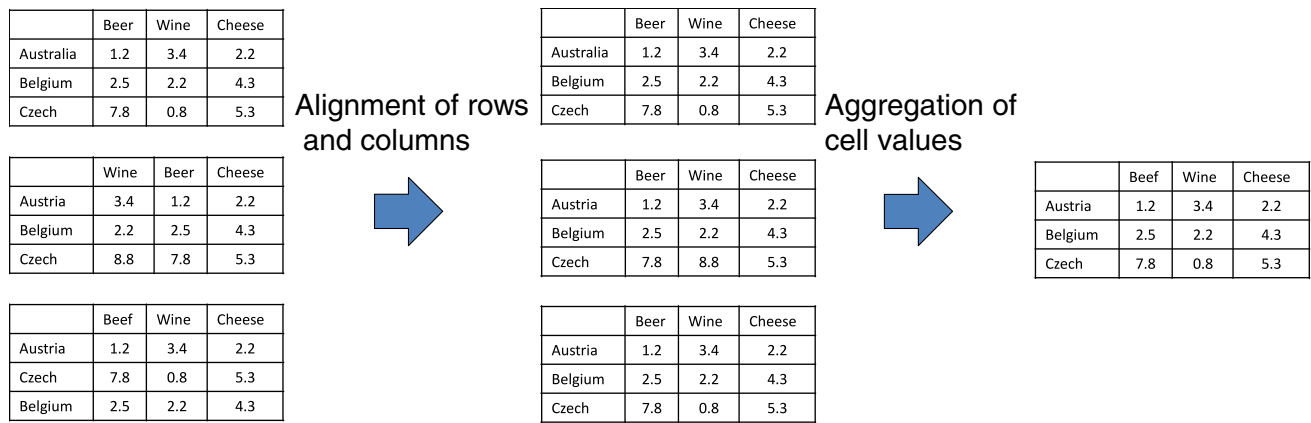[13] https://news.microsoft.com/bythenumbers/ms_numbers.

[14] http://www.lancers.jp.

	Beer	Wine	Cheese
Australia	1.2	3.4	2.2
Belgium	2.5	2.2	4.3
Czech	7.8	0.8	5.3

	Wine	Beer	Cheese
Austria	3.4	1.2	2.2
Belgium	2.2	2.5	4.3
Czech	8.8	7.8	5.3

	Beef	Wine	Cheese
Austria	1.2	3.4	2.2
Czech	7.8	0.8	5.3
Belgium	2.5	2.2	4.3

Alignment of rows and columns →

	Beer	Wine	Cheese
Australia	1.2	3.4	2.2
Belgium	2.5	2.2	4.3
Czech	7.8	0.8	5.3

	Beer	Wine	Cheese
Austria	1.2	3.4	2.2
Belgium	2.5	2.2	4.3
Czech	7.8	8.8	5.3

	Beer	Wine	Cheese
Austria	1.2	3.4	2.2
Belgium	2.5	2.2	4.3
Czech	7.8	0.8	5.3

Aggregation of cell values →

	Beef	Wine	Cheese
Austria	1.2	3.4	2.2
Belgium	2.5	2.2	4.3
Czech	7.8	0.8	5.3

Fig. 7 Example of aggregating tables created by three workers. After the rows and columns are aligned, the corresponding values from the three tables are integrated to obtain a single table

tasks such as image classification, using the majority criterion generally improves the accuracy of the final result.

In our case, however, the worker outputs are tables, which are complex objects composed of several headers and many values, so a simple majority criterion cannot be applied. We thus developed a two-stage algorithm for integrating multiple tables made by different workers for the same chart image. First, the rows and columns in the different tables are aligned, and then the values in corresponding cells are aggregated. Figure 7 shows an example of aggregating the tables created by three workers.

4.1 Alignment of rows and columns

In general, the order of rows and columns in a table is arbitrary (except obvious cases in which the order is naturally defined such as for dates), so different workers may order the rows and columns differently. For example, different workers may arrange the row labels ("London," "New York," "Tokyo") in the chart in Fig. 4 differently. Therefore, when aggregating the tables created by different workers, we first have to align the rows and columns among the different tables.

The names of rows (or columns) are the most important clue for judging whether two rows (columns) are identical; however, the names may contain errors or are sometimes missing in tables created by crowd workers. In that case, if the values in the rows (columns) are the same, the rows (columns) can be judged to be the same. Therefore, we introduce the similarity of two rows (columns) considering both their names and values and use it to find matching between rows (columns).

The measure we introduce here for measuring the similarity between two rows made by different workers is based on the probability of disagreement between the row headers and between the row values. Assume two workers w and w' transcribe the same row in a table and produce

rows $(z^w, x_1^w, \ldots, x_N^w)$ and $(z^{w'}, x_1^{w'}, \ldots, x_N^{w'})$, respectively, where z represents the header and x_n represents a row value. (We assume the two rows have the same number of values but later explain how we deal with rows with different numbers.) We assume that with probability α_z the two workers use the same label for the header and that with probability $1 - \alpha_z$ they use different labels. We also assume that the probability that two corresponding values are the same is α_x. These probabilities give the probability of having the two rows transcribed by the two workers:

$$\alpha_z^{I(z^w = z^{w'})} (1 - \alpha_z)^{I(z^w \neq z^{w'})} \prod_{x_n^w = x_n^{w'}} \alpha_x \prod_{x_n^w \neq x_n^{w'}} (1 - \alpha_x),$$

where I is a function that returns 1 when the condition holds and 0 otherwise. Using this probability, we define the similarity measure between rows transcribed by different workers as

$$I(z^w = z^{w'}) \ln \alpha_z + I(z^w \neq z^{w'}) \ln(1 - \alpha_z)$$
$$+ \sum_{x_n^w = x_n^{w'}} \ln \alpha_x + \sum_{x_n^w \neq x_n^{w'}} \ln(1 - \alpha_x).$$

We define the similarity between columns in the same way.

Using this similarity measure and the following procedure, we align the rows and columns in the tables created by the two workers.

1. Calculate the similarities between all rows produced by the two workers.
2. Assume that row pairs with high similarity contain identical data, and pair them up.
3. Pair any remaining unmatched rows with dummy rows.

A similar procedure is applied to columns. The alignment can begin with either rows or columns. In many cases, a

`Series` represents a time series and, in such cases, the `XValues` represent time or date. For such cases, there is usually less disagreement on the order of the `XValues` among workers' tables than there is on the orders of the `Series` objects. Therefore, starting with row alignment is a useful convention if the `Series` are arranged in a row (that is, the `Name` of a `Series` is used as the row header, and the `XValues` are used as the column headers). Note that changing the order of the columns changes the similarity between rows and vice versa; therefore, we should order rows and columns simultaneously. However, for simplicity, we assume they are independent and order the rows first and then order the columns. If three or more tables are to be aligned, one is chosen as a reference table, and the remaining tables are aligned with respect to the reference table.

4.2 Aggregating table headers and cell values

Since the results produced by crowdsourcing may contain errors, after the rows and columns of the tables are matched, the corresponding values from the tables are integrated to obtain a single table. The majority criterion is used to determine the final values for table headers since nominal values such as names are frequently used. The median value is used for cell values since numerical values are frequently used and the median is more robust to outliers than the average. Considering human errors as outliers rather than noise (as they are in instrument measurements) is appropriate because crowd workers can make errors of great magnitude. For example, consider a case in which the inputs from three workers are 1982, 1892, and 1982 and the actual value is 1982; the median matches the actual value while the average value greatly differs.

5 Case study

In this section, we describe our experiments on digitizing actual statistical chart images by using crowdsourcing. Our goal was to evaluate the feasibility of our framework and the effectiveness of the quality control mechanism.

5.1 Dataset and software

We evaluated our proposed framework and quality control mechanism experimentally by using chart images from the 2013 White Paper on Tourism[15] published by the Japan Tourism Agency. The white paper is published under a Creative Commons CC-BY license, and most of the statistical data are provided as figures embedded in HTML documents or PDF files, i.e., as one-star open data in Berners-Lee's

---

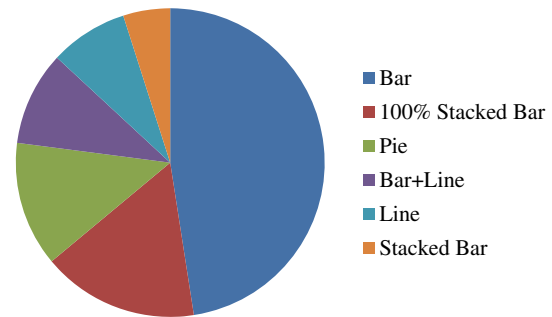[15] http://www.mlit.go.jp/statistics/file000008.html.

Fig. 8 Percentages for different types of charts in the dataset

scheme. Among the 104 images used in the white paper, 61 explicitly show values as data labels, and we used them as the gold standard for evaluating the correctness of the extracted values.

The dataset contains various types of charts. We categorized them into six categories: (a) bar chart, (b) line chart, (c) pie chart, (d) stacked bar chart, (e) 100 % stacked bar chart, and (f) bar+line chart. The percentages for these chart types in the dataset are shown in Fig. 8. Although the bar charts account for about half the dataset, the other chart types are also prevalent.

We compared the results for two crowdsourcing tasks. One simply asked workers to extract data from charts and put them in a spreadsheet ("Create Table" tasks), and the other asked workers to reproduce charts in a spreadsheet ("Reproduce Chart" tasks). We asked five workers to complete each task. We used the Lancers crowdsourcing service and paid 200 JPY (approximately 2 dollars) for each completed task. A worker was not required to do all tasks—23 different workers did at least one Create Table task, and 30 different workers did at least one Reproduce Chart task.

We implemented the table aggregation software using VB.NET. This software took the Excel files created by the workers as input and searched each file for a `Chart` object. If multiple `Chart` objects were found, it used only the one with the first index number. For each `Series` object in the `Chart` object, it extracted the values of the `Series.Name`, `Series.XValues`, and `Series.Values` properties from the corresponding cells in the worksheet as the row headers, column headers, and cell values, respectively. The table aggregation algorithm was then applied to the set of worker tables (with $\alpha_x = \alpha_z = 0.9$), and the aggregated table was stored as a worksheet in a new Excel file.

5.2 Accuracy of worker tables

We manually made gold standards for the transcribed tables and evaluated the accuracy of the tables created by the crowd workers. We separately evaluated the accuracies of the row and column headers and the cell values. Figure 9 shows the
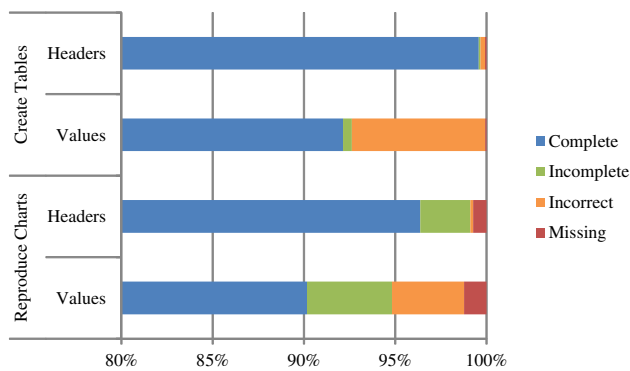
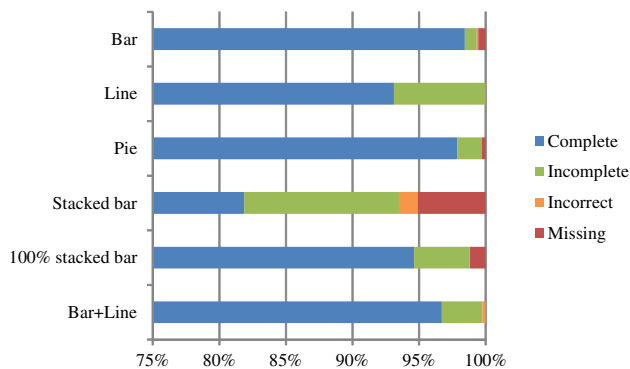Fig. 9 Percentages for different types of errors in worker tables

Fig. 10 Errors in table headers in worker tables for different types of charts

Fig. 11 Errors in cell values in worker tables for different types of charts

percentages of different types of errored cells for both tasks. "Incomplete" means some data values were not exactly the same as the gold standard, such as different spelling or values without appropriate units. "Incorrect" means that the values were simply incorrect mainly due to mistyping or another mistake, and "Missing" means some data values were missing from the table. Although the dataset contained complex charts such as the ones shown in Fig. 1, both tasks resulted in accuracies higher than 90 % for both table headers and cell values, which indicates that our approach using crowdsourcing is promising. The Reproduce Chart task resulted in fewer incorrect header and cell values than the Create Table task. This might be because the reproduced charts made it easier for the workers to spot errors. On the other hand, the Reproduce Chart task resulted in more incomplete and missing values.

Figures 10 and 11 show the percentages of errors in table headers and cell values generated by the Reproduce Chart task for different types of charts. Most "incomplete" headers were missing units of measure. The pie charts had many missing values. Pie charts usually display percentages as well as numeric values, but many workers did not transcribe them into their tables but instead calculated them from the numeric values using a function of Excel. For example, the pie chart in Fig. 12 shows percentages, but they are not found in the
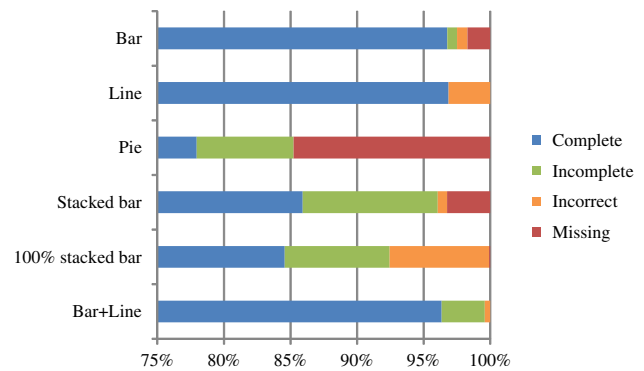
table. They are displayed by specifying data label formats. The totals for the stacked bar chart are also missing for the same reason. Although we counted them as "missing values" in our evaluation, they can be recovered from the numeric values in the table and thus should not cause major problems in practice.

5.3 Accuracies of aggregated tables

We next measured the accuracies of the aggregated tables. At least three tables are required so that using the majority criterion or the median works well. We compared two different settings. In one, all five worker-generated tables were used for each chart image; in the other, three randomly selected tables were used for each image. Figure 13 shows the percentages for different types of errors after aggregation. Aggregation greatly improved the accuracy for cell values. It also eliminated most of the incorrect and missing headers, but it was not very effective for reducing the incomplete headers.

Figures 14 and 15 show the percentages of errors in the aggregated tables from five workers for different types of charts. Most of the incomplete headers were due to lack of appropriate units. Many workers did not write them in the cells, so the majority criterion did not work well. Although we could recover some missing "percentages" by retrieving cell style information, a more general handling of missing units is left for future work. Most missing cell values in the stacked bar charts were the totals of the stacked values, which can be recovered, as explained in the previous subsection.

6 Analysis of results when workers used chart digitizing software

As discussed in Sect. 2, there are various types of chart digitizing software available for digitizing chart images in documents such as research papers. Such software is useful for obtaining the values of data points in a graph, but it
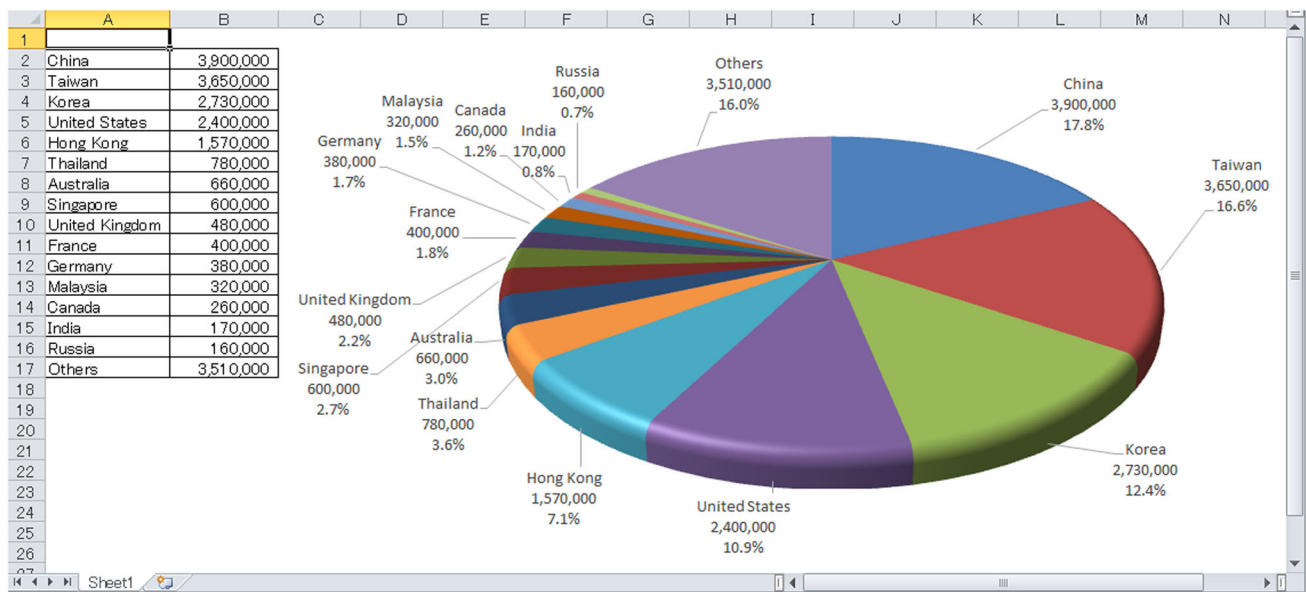
Fig. 12 Example of pie chart and corresponding table. Although pie chart displays percentages, they are not shown in table
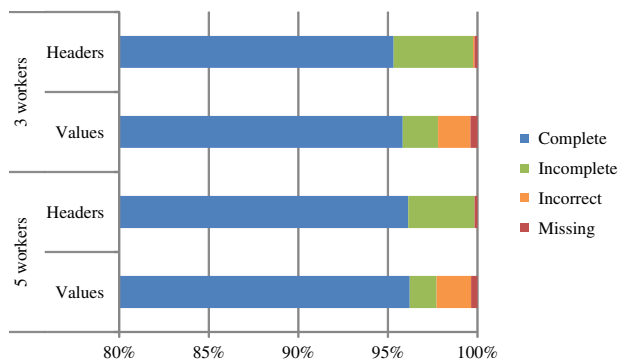

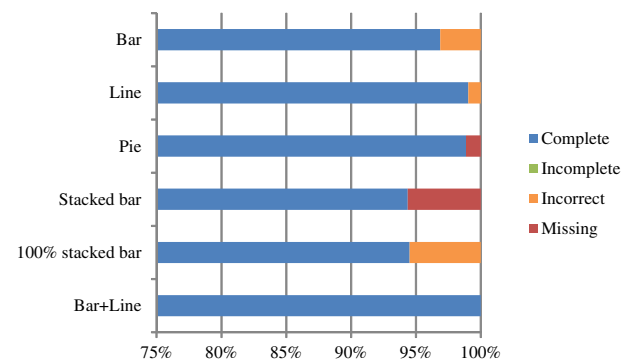**Fig. 13** Percentages for different types of errors in aggregated tables

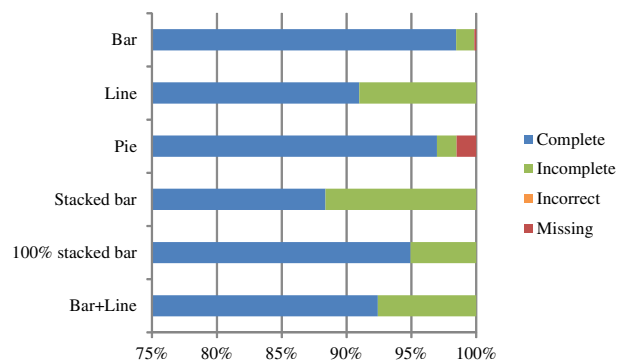**Fig. 14** Errors in table headers in aggregated tables for different types of charts

Fig. 15 Errors in cell values in aggregated tables for different types of charts

is not fully automatic and thus needs human intervention. Our framework does not prevent workers from using such software. In this section, we describe the experiments we conducted to analyze the effect of using it. In these experi-ments, we recommended to the workers that they use chart digitizing software if they thought it would be useful and asked them, if they used such software, to identify the soft-ware used.

6.1 Example of chart digitizing software

Chart digitizing software is especially useful for extracting data from charts without data labels, like the one shown in Fig. 16, from which workers need to estimate the values by eye. Chart digitizing software can determine the values of the data points specified by the user once the user has calibrated the axes of the graph, as in the following steps:

– First, the user is asked to click two reference points on the X-axis and two reference points on the Y-axis.

Fig. 16 Example of chart without values as data labels in 2013 White Paper on Tourism. Workers needed to estimate values by eye
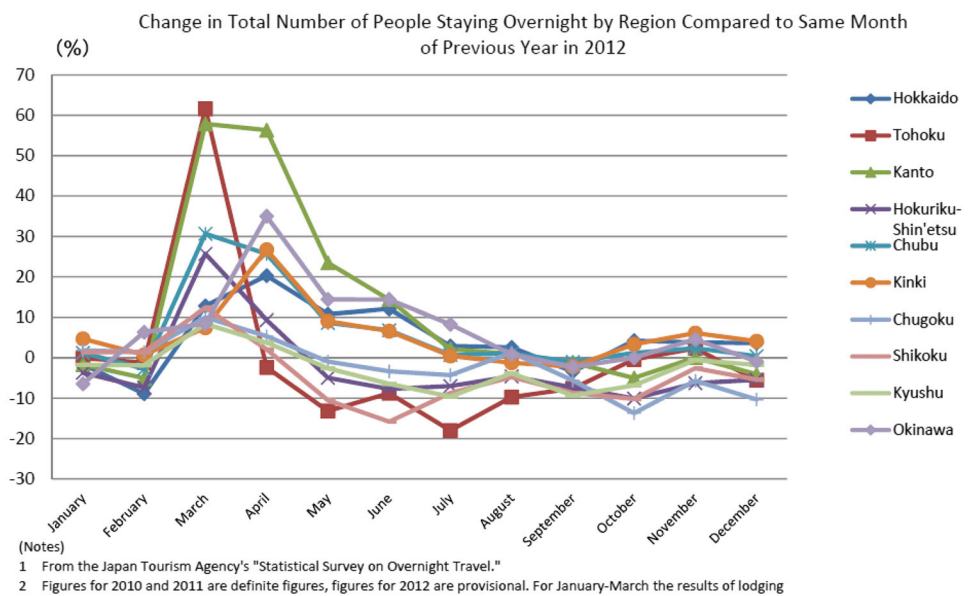
– Then the user is asked to enter the *X*-values of the points on the *X*-axis and the *Y*-values of the points on the *Y*-axis.
– The software calculates the scales of the X- and Y-axes by using the values of the reference points.
– In "manual" mode, the user is asked to specify each point on each line, and the software calculates the $X–Y$ values of the point.
– In "automatic" mode, the user specifies the color of the line in the chart, and the software automatically calculate the $X–Y$ values of points on the line.

Figure 17 shows a screenshot of axis calibration in WebPlot-Digitizer. Note that automatic mode requires axis calibration as well as color specification, so it is not truly automatic and needs human intervention.

6.2 Experimental settings

In the crowdsourcing tasks for the chart digitizing experiments, we recommended to the workers that they use chart digitizing software if they thought it would be useful, and we provided links to the following chart digitizing software as examples.

– WebPlotDigitizer,
– GSYS,[16]
– SimpleDigitizer,[17]
– G3DATA.[18]

After they completed each task, we asked them whether they had used chart digitizing software and, if so, which one. In addition to the software we recommended, some users reported that they used the following software.

– Graphcel,[19]
– !0_0! Excel.[20]

As chart images to digitize, we chose 37 chart images without data labels from the 2011 White Paper on Information and Communications[21] published by the Ministry of Internal Affairs and Communications of Japan. Although there were no data labels in these charts, we found the original CSV files from which the charts were made. We used these CSV files as gold standards to evaluate the accuracy of the tables made by the workers. We used the Lancers crowdsourcing service and asked five workers to complete each task (reproduce a chart image as a chart in an Excel spreadsheet). We paid 200 JPY (approximately 2 dollars) for each completed task.

Figure 18 shows the percentages of the tasks completed using different types of chart digitizing software. GSYS and SimpleDigitizer were used the most, probably because detailed Japanese documentation was available for them. Note that 28 % of the tasks were completed without using any chart digitizing software.

For each set of tasks completed using specific software, we calculated the root mean squared logarithmic error (RMSLE):

[16] http://www.jcprg.org/gsys/.

[17] http://www.alrc.tottori-u.ac.jp/fujimaki/download/windows.html.

[18] http://www.frantz.fi/software/g3data.php.

[19] http://www.vector.co.jp/soft/win95/business/se247204.html.

[20] http://www.vector.co.jp/soft/win95/art/se312811.html.

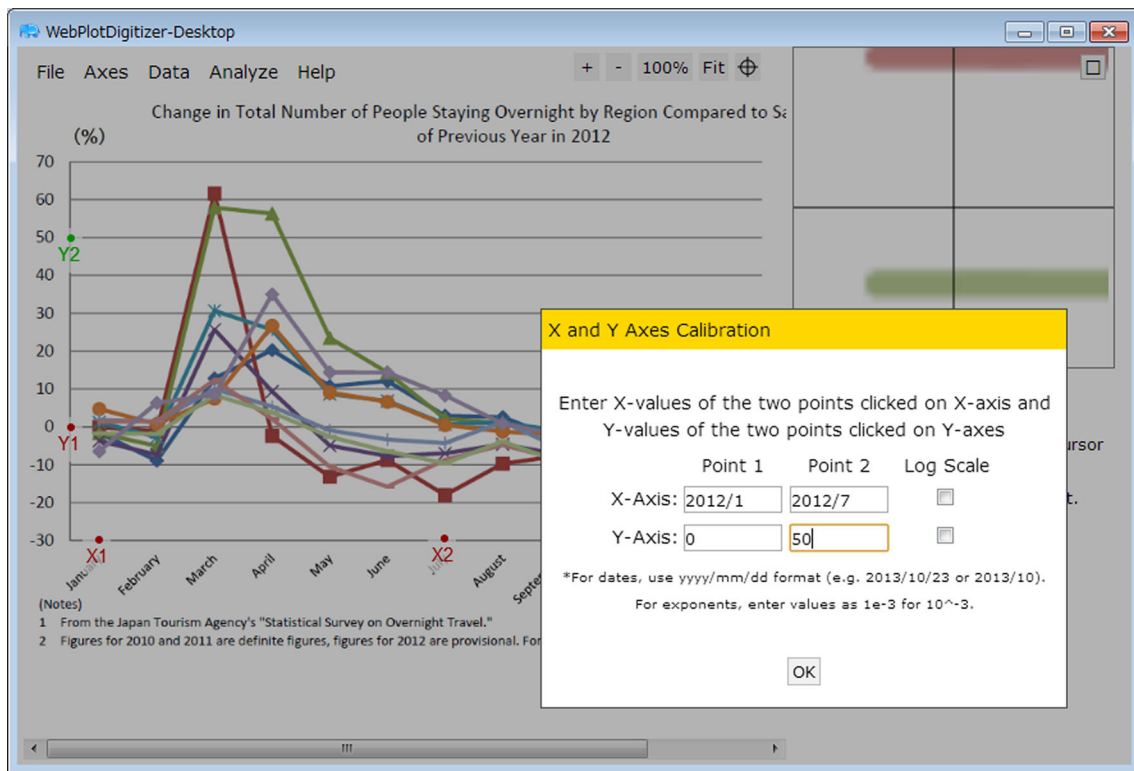[21] http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/index.html.

Fig. 17 Axis calibration in WebPlotDigitizer. User is asked to enter *XY* coordinate values of reference points
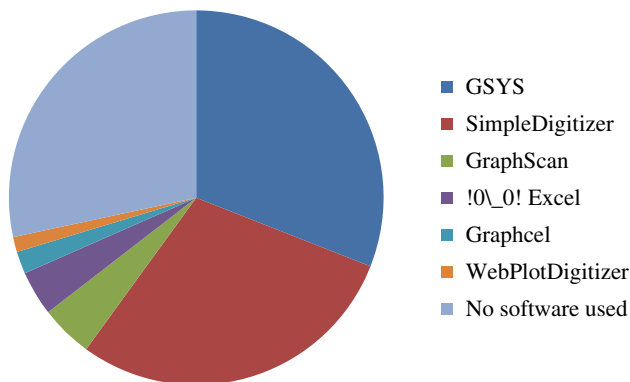

Fig. 18 Percentages of tasks completed using chart digitizing software by type of software

$$E = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\ln\left(x_i^{\text{worker}} + 1\right) - \ln\left(x_i^{\text{true}} + 1\right)\right)^2},$$

where N is the total number of cells in the set of tasks, x_i^{true} is the true value for cell i, and x_i^{worker} is the worker's value for cell i.

For the tasks for which there were results using software and results not using software, we calculated the error for each case. Figure 19 shows the error distributions using software and not using software. Each point represents a task.

Its *X*-value shows the error using software, and its *Y*-value shows the error not using software. We can see that using chart digitizing software did not always reduce the error.

Figure 20 shows the comparison of the errors among the different types of software used. Although the results for some types of software are not significant due to limited numbers of used tasks as shown in Fig. 18, there were significant differences among the more popular choices (GSYS, Simple Digitizer, and no software). Tasks completed using a certain software contained larger errors than tasks completed using another software and tasks completed without using any software. Note that these differences should not be attributed to the software itself. The users may not have fully understood how to use the software or simply lacked the skills or carefulness needed to complete the task accurately. These results do show that even if workers use chart digitizing software, errors caused by human operation are unavoidable and that quality control is still necessary to ensure accuracy.

7 Summary and future work

Converting legacy open data in, for example, image files into machine-readable format is an important step toward realizing the potential of the Linked Open Data cloud, but it is labor-intensive, and there have been few related research

Fig. 19 Distribution of errors using software and not using software
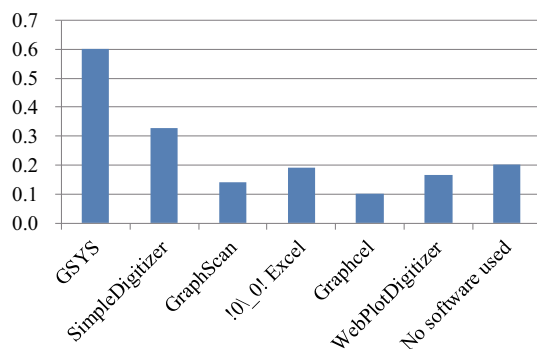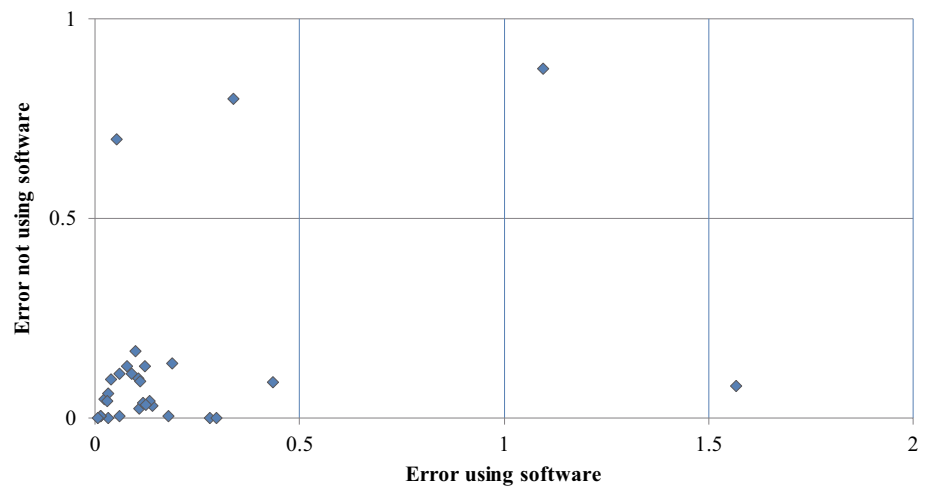

Fig. 20 Errors for different types of chart digitizing software

efforts. We proposed using crowdsourcing to digitize chart images and introduced a task design. In this design, crowd workers are asked to reproduce chart images as embedded chart objects in spreadsheets, which enables automatic identification of table data structures from the properties of the chart objects. To reduce the number of errors inherent in crowdsourced results, we developed a quality control mechanism. Multiple workers are asked to digitize the same chart image, and the tables they create are aggregated. Experimental results demonstrated that our approach is effective, but many tasks remain for future work. They can be grouped into four main areas, as described below.

7.1 Task design

Several lessons were drawn from the results of our experiments. The inconsistency in the use of units could be reduced by defining units as cell styles in advance and asking the workers to use them. In the experiments, we asked the workers to upload their Excel files to a file server, and we downloaded them manually to a local machine, on which we ran a table aggregation program. This procedure is prone to problems, such as lost and misidentified files. Performing all

the processes in a single cloud environment would greatly improve the throughput and accuracy of the chart digitizing tasks.

7.2 Table aggregation

We assumed that the workers had consistent abilities, and we used simple aggregation methods for table integration; however, workers in a crowdsourcing setting generally have various degrees of ability. Application of more sophisticated statistical quality control methods that consider worker-dependent abilities (e.g., [21,22]) is a possible future direction for improving integration accuracy.

7.3 Integrating chart digitizing software in the framework

As shown in Sect. sec:software, using chart digitizing software did not always reduce the error size, but if appropriately used, it could contribute to reducing it. The usefulness of software depends on the type of chart to be digitized, so recommending to users whether or not to use software for each task and, if so, recommending specific software could contribute to producing more accurate results. Different types of software cause different degrees of error, which may have different statistical patterns. This means that utilizing information on whether or not software was used and, if it was, what type of software was used should help reduce the error size in the final results.

7.4 Converting tables into RDF format

The next step according to the roadmap for the Linked Open Data cloud is converting tables into RDF format. The dimensions, measures, and attributes in the RDF Data Cube Vocabulary, a framework for representing statistical tables,

generally correspond to headers, values, and units of values in statistical tables. After the headers and values are extracted using a chart digitizing approach, we have to relate them to a formally defined vocabulary. This process is also difficult to perform solely by computer; it requires human intelligence. Therefore, the use of crowdsourcing to convert tables into RDF format is important future work. Collecting publicly available spreadsheets with charts and extracting names from them would help in constructing a vocabulary for describing statistical data.

7.5 Structured data extraction through visualization

We reproduced image-formatted charts in spreadsheets to enable us to extract table-formatted data from them. However, there are many other data types that are not provided in visualized formats such as CSV-like texts and spreadsheets without charts. Producing charts from these non-visualized data would make the data easier to understand; moreover, such processes would help in extracting the structures of the data as a by-product. This is referred to as *unsupervised* visualization of the data while chart reproduction from images is referred to as *supervised* visualization.

Compliance with ethical standards

Conflicts of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Oyama, S., Baba, Y., Ohmukai, I., Dokoshi, H., Kashima, H.: From one star to three stars: upgrading legacy open data using crowdsourcing. In: Proceeding of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA 2015) (2015)
2. Shadbolt, N., O'Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H., Hall, W., schraefel, m. c.: Linked open government data: lessons from data.gov.uk. IEEE Intell. Syst. **27**(3), 16 (2012)
3. Law, E., von Ahn, L.: Human Computation. Morgan & Claypool Publishers, San Rafael (2011)
4. Hosseini, M., Phalp, K., Taylor, J., Ali, R.: The four pillars of crowdsourcing: a reference model. In: Proceedings of the 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS 2014), pp. 1–12 (2014)
5. Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: RDF123: from spreadsheets to RDF. In: Proceedings of the 7th International Semantic Web Conference (ISWC 2008), pp. 451–466 (2008)
6. Mulwad, V., Finin, T., Joshi, A.: Automatically generating government linked data from tables. In: Working Notes of AAAI Fall Symposium on Open Government Knowledge: AI Opportunities and Challenges (2011)
7. Bizer, C.: The emerging web of linked data. IEEE Intell. Syst. **24**(5), 87 (2009)
8. Meroño-Peñuela, A., Hoekstra, R., Scharnhorst, A., Guéret, C., Ashkpour, A.: Longitudinal queries over linked census data. In: Proceedings of the 10th Extended Semantic Web Conference (ESWC 2013), pp. 306–307 (2013)
9. Kalampokis, E., Tambouris, E., Tarabanis, K.: Linked open government data analytics. In: Proceedings of the IFIP Electronic Government Conference (EGOV 2013), pp. 99–110 (2013)
10. von Ahn, L., Dabbish, L.: Designing games with a purpose. Commun. ACM **51**(8), 58 (2008)
11. Cao, L.: Metasynthetic Computing and Engineering of Complex Systems. Springer, Berlin (2015)
12. von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: human-based character recognition via web security measures. Science **321**(5895), 1465 (2008)
13. Quinn, A.J., Bederson, B.B.: AskSheet: efficient human computation for decision making with spreadsheets. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW 2014), pp. 1456–1466 (2014)
14. Fan, J., Lu, M., Ooi, B.C., Tan, W.C., Zhang, M.: A hybrid machine-crowdsourcing system for matching web tables. In: Proceedings of the 30th IEEE 30th International Conference on Data Engineering (ICDE 2014), pp. 976–987 (2014)
15. Ermilov, I., Auer, S., Stadler, C.: User-driven semantic mapping of tabular data. In: Proceedings of the 9th International Conference on Semantic Systems (I-SEMANTICS 2013), pp. 105–112 (2013)
16. Simperl, E., Norton, B., Vrandečić, D.: Crowdsourcing tasks in linked data management. In: Proceedings of the 2nd Workshop on Consuming Linked Data (COLD 2011) (2011)
17. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st International Conference on World Wide Web (WWW 2012), pp. 469–478 (2012)
18. Noy, N.F., Mortensen, J., Musen, M.A., Alexander, P.R.: Mechanical turk as an ontology engineer?: using microtasks as a component of an ontology-engineering workflow. In: Proceedings of the 5th Annual ACM Web Science Conference (WebSci 2013), pp. 262–271 (2013)
19. Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., Lehmann, J.: Crowdsourcing linked data quality assessment. In: Proceedings of the 12th International Semantic Web Conference (ISWC 2013), pp. 260–276 (2013)
20. DiFranzo, D., Hendler, J.: The semantic web and the next generation of human computation. In: Handbook of Human Computation, pp. 523–530. Springer, New York (2013)
21. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 20–28 (1979)
22. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: Advances in Neural Information Processing Systems 22 (2009)