Making Legacy Open Data Machine Readable by Crowdsourcing*

Satoshi Oyama Hokkaido University
Yukino Baba Kyoto University
Ikki Ohmukai NII
Hiroaki Dokoshi Hokkaido University
Hisashi Kashima Kyoto University

Abstract

An approach is described for converting legacy statistical data in image format into a machine-readable and reusable format by using crowdsourcing. Requesting crowd workers not only to extract tables from graph images but also to reconstruct them in spreadsheets can produces structures including attribute names and values as properties of the reconstructed graph objects. A quality control mechanism was developed 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. Experimental results using the White Paper on Tourism published by the Japan Tourism Agency demonstrated that the proposed approach is effective.

Introduction

The most prominent of the recent open data initiatives to publish various kinds of data in electronic format are the ones for statistical data gathered by governmental agencies (Shadbolt et al. 2012). However, a significant percentage of such statistical data is published as charts or graphs in image or PDF files, which are not suitable for automatic processing by machine. 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 have been developed. However, such software is designed for manual use and requires human intervention, such as in calibrating the chart axes, making it unsuitable for automatically extracting data from a large number of data charts. Since data charts are designed to help people better understand data, people are better at understanding them than computers. We have thus taken a human computation approach to chart digitizing: use crowdsourcing to extract structured data from charts in legacy file formats such as image and PDF files.

*The full version of this paper appeared in the Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA 2015) (Oyama et al. 2015).

Task Design for Chart Digitizing

Our goal is to accurately extract data from charts without data omission or modification. A naive task design for crowdsourced chart digitizing is asking workers to simply read data from a chart and write them in a table in CSV or spreadsheet format. Instead, we ask workers to visually reproduce a chart image as a chart object in a spreadsheet using the functions of spreadsheet software. This enables the requester to obtain a table linked to a chart object representing the data in the table. Such a data structure is essential for controlling the quality of the digitizing work; it provides an efficient way to aggregate tables made by different workers, and enables us to follow the common practice of quality control in crowdsourcing that asks multiple workers to complete the same task and aggregate the results. On the other hand, it is relatively complicated to integrate multiple tables in CSV or spreadsheet formats, which do not provide a data structure.

Structured Data Extraction through Visualization

During the process of visually reproducing a chart image, the worker has to specify the properties of the chart object in the spreadsheet that reflect the structure of the data represented in the chart. Such properties can be accessed by using a computer program using an application programming interface. For example, Microsoft Excel uses the following data representation: (1) a `Chart` has one or more Series, (2) each `Series` has a `Series Name`, and (3) each `Series` has `XValues` and `Values`. This data representation is common among different types of graphs. `Series Names` and `XValues` correspond to the row and column headers of a table. 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.

The data representation above is independent of the choice of rows and columns in representing data in a table. A worker representing chart data in a table can arrange the data series either in rows or in columns, but in either case the data series are represented by a `Series`. These characteristics are extremely useful in integrating multiple tables.

They are also useful in representing data using semantically richer formats, such as RDF with the DataCube Vocabulary.

Aggregating Multiple Tables

Tables made by different workers are integrated by first aligning the rows and columns among the tables and then determining the cell values from the values in the tables being integrated. The first step is necessary because in general the order of rows and columns in a table is arbitrary, and different workers may give rows and columns in different orders. 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. We introduce the similarity of two rows (columns) considering both their names and values and use it to find matching between rows (columns). The similarity measure between two rows (columns) made by different workers is based on the probability of disagreement between the row (columns) headers and between the row (columns) values. Using this similarity measure, we align the rows and columns in the tables created by the two workers. In the second step, for nominal values such as row/column headers, we use majority voting to aggregate the values of the different workers. For numerical values, typically item values in tables, we use the median rather than average since the majority of errors in chart digitizing are outliers, such as mistaking 100 as 1000, and the median is more robust against outliers. For more details of the table aggregation algorithm, see Oyama et al. (2015).

Evaluation

We evaluated our proposed approach experimentally by using chart images from the 2013 White Paper on Tourism published by the Japan Tourism Agency. 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. We compared the results of two different 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 used the Lancers crowdsourcing service. Figure 1 shows the percentages of different types of error 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, 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. For example, pie charts usually display percentages

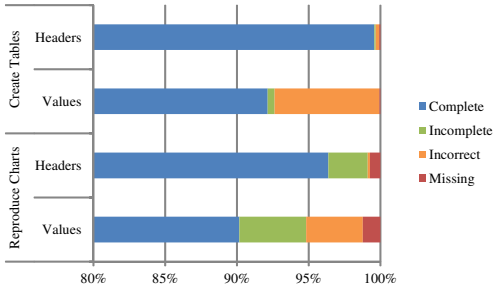

Figure 1: Percentages for different types of errors in worker tables

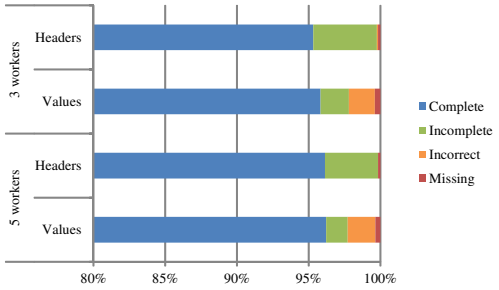

Figure 2: Percentages for different types of errors in aggregated tables

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. 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. Figure 2 shows the percentages for different types of errors after table aggregation. Aggregation greatly improved the accuracy for cell values. It also eliminated most of the incorrect and missing headers while it was not very effective for reducing the incomplete headers. 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 part of our future work. For more detailed discussion on the experimental results, see Oyama et al. (2015).

References

Oyama, S.; Baba, Y.; Ohmukai, I.; Dokoshi, H.; and Kashima, H. 2015. From One Star to Three Stars: Upgrading Legacy Open Data Using Crowdsourcing. In *DSAA 2015*.

Shadbolt, N.; O'Hara, K.; Berners-Lee, T.; Gibbins, N.; Glaser, H.; Hall, W.; and m. c. schraefel. 2012. Linked Open Government Data: Lessons from Data.gov.uk. *IEEE Intell Syst* 27(3):16–24.