Query-Biased Summaries for Tabular Data

Vincent Au*
Australian National University
vincent.au@anu.edu.au

Paul Thomas*
Microsoft
pathom@microsoft.com

Gaya K. Jayasinghe
Data61, CSIRO
gaya.jayasinghe@csiro.au

ABSTRACT

Government, research, and academic data portals publish a large amount of public data, but present tools make discovery difficult. In particular, search results do not support a user's decision whether or not to commit to a download of what might be a large data set.

We describe a method for producing query-biased summaries of tabular data, which aims to support a user's download decision—or even to answer the question on the spot, with no further interaction. The method infers simple types in the data and query; automatically refines queries, where that makes sense; extracts relevant subsets of the complete table; and generates both graphical and tabular summaries of what remains. A small-scale user study suggests this both helps users identify useful results (fewer false negatives), and reduces wasted downloads (fewer false positives).

Keywords

data portals; tables; information retrieval

1. SUMMARIES FOR TABLES

An increasing amount of data is being published online, by research bodies (e.g. ands.org.au), governments (data.gov.uk, data.gov), and third-party brokers (govpond.org). Data portals, which make such data available, typically include search facilities to help visitors find data they need.

Search results are represented by summaries, which describe each retrieved data set. A good summary captures the relevance of a linked document, helping a user to decide whether or not to investigate further; an especially good summary may even answer a user's question without any further interaction. In some cases (e.g. data.gov) these summaries are *static*, in that they are the same regardless of a user's query. Alternatively, *query-biased summaries* such as those at data.gov.hk are modified to take into account a user's query [5]. Query-biased summaries can lead to higher precision and recall in identifying relevant documents, as well as increased speed and reduced clickthrough [2, 5, 6].

*Work carried out at CSIRO.

ADCS '16, December 05 - 07, 2016, Caulfield, VIC, Australia

ISBN 978-1-4503-4865-2/16/12...$15.00

DOI: http://dx.doi.org/10.1145/3015022.3015027

However, the techniques for query-biased summaries of text are not appropriate for other forms of data. Figure 1 illustrates a data search with query-biased summaries—in this case at the Australian Bureau of Statistics, but in a form common to the majority of portals worldwide. The portal does in fact contain the data needed, but despite the tailored summaries it is not clear whether the listed data sets are relevant: the first result talks about female workers, but may not mention pay; the second talks about pay, but perhaps not sex; in either case, the searcher must commit to downloading the entire data set just to check. Further, since the summary is restricted to text, it is not possible to show parts of the data set itself. Searches in other data stores—for example, local folders or shared drives of spreadsheets—have similar problems.

In this work, we build a mechanism for query-biased tables of categorical or numeric data, to support a user's download decisions and to provide instant answers where possible.

Context. In doing this, we focus especially on the needs of medium to large-scale publishers of heterogenous data. If a data set is small, it is relatively simple to add appropriate metadata—for example, complete textual descriptions—which may or may not help ranking but will certainly support the download decision. This manual effort can also extend to writing summaries which will answer common questions from the data. If data is large but homogeneous, for example if the same data is collected and published on a regular basis, the summarisation task is one of pointing out differences between sets (e.g. publication dates) and again this is fairly trivial. However, organisations such as government agencies, large organisations such as universities, and aggregators cannot rely on manual annotation—their data sets are too large—nor on simple metadata operations—their data is too varied. In these instances we would like a system driven entirely from the data, and not tied to any one domain or query type.

Related work. Systems such as web search engines or Wolfram Alpha can provide answers or graphical summaries of data in their collections. These systems however rely heavily on manual curation and a unified semantics (see e.g. http://www.wolfram.com/knowledgebase/), to drive both query parsing and summary generation. This approach is clearly powerful, but most data portals do not have the resources to mark up data this way and markup provided by individual authors may be inconsistent, unreliable, or simply absent. The approach is even less appropriate for less-specialised, medium-scale systems such as institutional or personal collections.

Various automatic data visualisation tools exist for data exploration and visualisation, including Tableau, VizDeck, Qlik, Power BI, and many others (tableau.com, vizdeck.com, qlik.com, powerbi.microsoft.com). These systems typically identify the type of data—

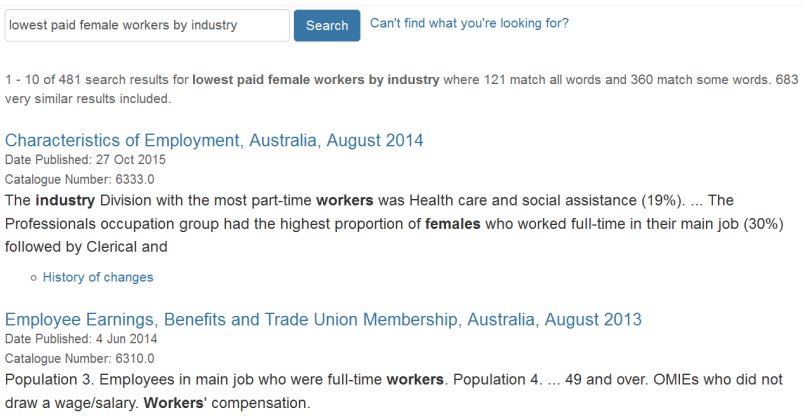

Figure 1: Retrieval results for the query "lowest paid female workers by industry", at the Australian Bureau of Statistics (May 2016). The summaries do not provide actual data; nor do they make it clear which, if any, data sets would be worth downloading.

nominal, currency, location, etc.—and suggest compatible visualisations. More advanced systems can rank possible visualisations and make suggestions based on data anomalies or visual interest [1, 3].

These data exploration tools provide a good deal of control but require a user to identify not just a data set, but the subsets (e.g. columns) and questions which are of interest. By contrast, we are interested in helping users decide which data sets are useful in the first place, by automatically identifying and calling attention to those parts relevant to a query.

2. METHOD

We concentrate on the most common data format, which is also that with the least metadata support: simple tables such as those in spreadsheets. These are extremely common: for example we estimate there are 2000 such sets in ands.org.au, 141k in .gov.au, 200k in data.gov.uk, and 1.6M in .gov, and millions available in the public web on top of unknowable numbers in private collections. Each table is allowed to have any number of rows and columns; a single value per cell; and a single, optional, description.

To rank these tabular data sets with respect to a query, we use the ranked retrieval algorithm for tabular data proposed in our earlier work [4]. For each ranked data table, we produce a summary with (1) a tabular extract, and (2, when appropriate) a graphical summary.

Matching cells. To generate a query-biased summary, we need a subset of cells relevant to the query. One way to extract such a subset is to extract the intersection of rows and columns (along with their headings) that best answers the query.

For example, given the query "indigenous population Melbourne 1970s" we can infer that the user is interested in the population statistics of indigenous people in the city of Melbourne (in the state of Victoria, in Australia) for the years from 1970–1979. This implies that we are looking for columns or rows where the heading contains the terms "indigenous" or "population", columns or rows (ideally containing place names) with cell values "Melbourne" or synonyms (e.g. "MEL"), and columns or rows (ideally time) where cell values are from 1970 to 1979.

Therefore, prior to matching query terms with a subset of columns and rows in the table, we infer data types for each column and normalise data values according to a simple convention. For each query term, we infer data types and normalise using the same method. New data types can be easily recognised and normalised with simple algorithms, but we currently use six of the most common:

Dates are recognised in a variety of formats and stored with the same precision as in the source table, year first: e.g. "March 1954" is translated to "195403". Dates in queries are similarly translated and matched as prefixes, so the query "1954" would match this example but "apr 1954", translated to "195404*", would not. We also recognise date ranges in the query, as "1990s" or "1985-1987".

Numbers are stored stripped of thousands separators, and shifted according to any SI suffix such as "M" or "k". **Rates** marked with e.g. "%" or "ppm" are also recognised, as are **currencies**.

Place names are identified from a gazeteer and are stored as a single token, fully qualified to include surrounding areas. For example, "Sydney" is translated to "australia_newsouthwales_sydney". Placenames in queries are again treated as prefixes, so the query "Australia" matches the value "Sydney" but not vice-versa.

Any other value is classed a **string** and kept as-is.

Multiple types may be inferred for each term. For example, the term "1990" may be of data types date, numeric, or string, and the term "Sydney" may be a place name or string. For each column in a table, we inspect the types for the first twenty rows, less any missing values, and select the most constrained common type. For query terms we use the most constrained matching type. We used a similar strategy of inference and normalisation in earlier work [4], when indexing and ranking tables with respect to a query.

Inferring missing column headings. Column headings are arguably the most important cells in a table, as they provide context to the data. Where a heading is missing, inferring one can help determine relevance and help the final summary. For example, a table may have a column with containing cell values "England", "Ireland", "New Zealand", etc., which could be titled "Country" and matched against "country" or "countries" in a query; the column title could also appear in a summary.

We infer missing column headings as follows. First, the columns with missing headings are compared with the columns with headings in other data tables. If the highest cosine similarity is above a threshold $\alpha = 0.8$ (chosen empirically), we use the same heading.

Second, when the cosine similarity between the two column vectors is less than α but greater than a threshold $\beta = 0.4$, we consider the column heading a near match. The description for the data table often contain clues to infer missing column headings. Therefore, we form a heading from the intersection of terms in the description and terms in near matches.

Finally, if the column heading is not inferred in the above two steps, we take the intersection of terms appearing more than three

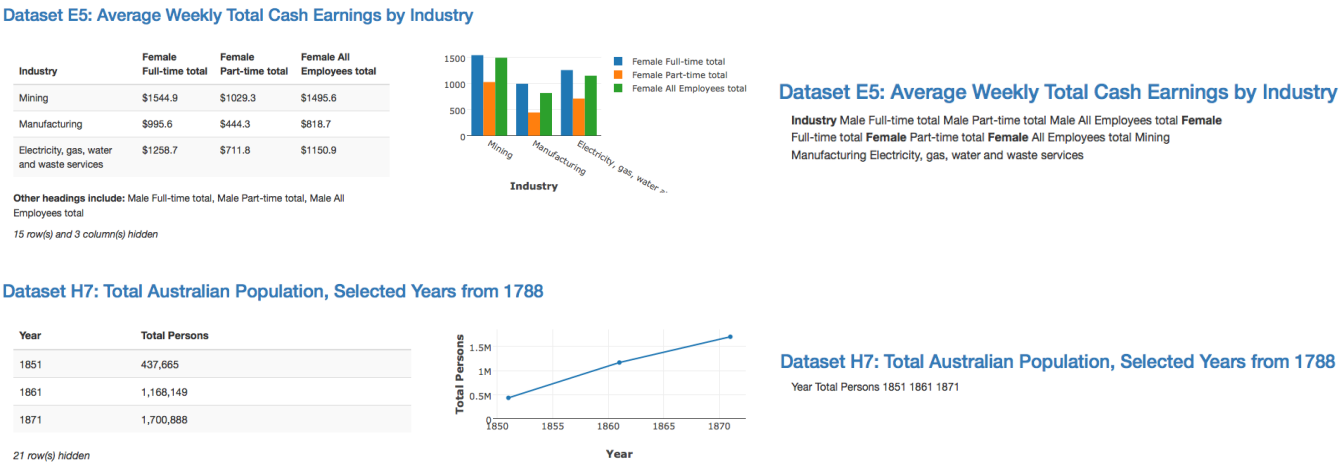

Figure 2: Query-biased summaries for queries "lowest paid female workers by industry" (top) and "population in Australia from 1850–1875" (bottom). Our tabular and graphical summaries are at left, text summaries at right.

times in the column and terms appearing in the description. The intuition here is that frequent terms in a column may contain clues to what the column heading is. For example, given cell values "state taxes", "federal taxes", etc., "taxes" may be an appropriate heading.

Generating tabular summaries. We use several heuristics to produce tabular summaries. First, at least for data tables written in English, the importance of a column in providing context to data often decreases from left to right. For example, a data table about unemployment rate in different states may have "State" as the first column. Therefore, we always include the first column in the tabular summaries. Second, we include columns where the data type and the terms in the heading match with the data type and the terms in the query. Third, we include rows and columns where column type and cell value match with any of the query term type and value. Fourth, when a data table contains columns of dates but no date is specified in the query, we assume that the user is interested in the most recent data, and mark those rows as relevant to the query. Fifth, we include date columns when the query contains a date if no date column has been included before. Sixth, we add columns in which the ratio of distinct cell values to total number of rows is less than 0.1. Here the intuition is that these columns provide context to the data based on a common categorisation (e.g., a column with distinct values "male" and "female"). Finally, we mark the first and last numeric columns of the data table as relevant if no numeric column have been included in previous steps. This is because they often are dependent variables showing outcome for combination of inputs in other columns.

After applying these heuristics we have a list of columns and rows relevant to the query. When we have identified fewer than four columns, we add the first four columns to the summary even if they were not otherwise relevant. Regardless of how many rows and columns are identified, at most three rows and five columns are shown in the final summary for succinctness.

Generating graphical summaries. Again, we use several heuristics to produce graphical summaries. The current implementation produces three types of graphical summaries that address most of the common scenarios in our dataset.

1. When a date column and a numeric column are selected as relevant, we produce a time series plot taking the former as the x axis, and the latter as the y axis. We use the respective column headings for axis titles.

2. For a table where the first column is of strings with a unique value in each row; one other selected column headings are dates; and other cell values are of type numeric, we produce a time series plot using a distinct colour for each selected row. The x axis is named for the column of dates, and the colour assigned to each row is shown in a legend.

3. When the first column is of strings with unique values in each row; and numeric columns are selected as relevant to the query, we produce a bar chart. Bars are grouped according to the string, with a different colour for each numeric column. When more than one numeric column is selected, a legend explains them on the graph.

We ignore all other cases, and produce only the sub-table.

We augment the snippet with the number of columns and rows hidden, and a list of column headings not shown in the summary.

The left side of Figure 2 shows two query-biased summaries generated for data tables for the queries "lowest paid female workers by industry" and "population in Australia from 1850–1875". In each case, the summary presents salient subsets of the data, both as an extract from the table and as a plot. This either provides the answer immediately, or provides evidence of the utility of the full table. Note that in neither case is this apparent from the table title alone.

3. USER STUDY

We performed a small user study to assess (1) effectiveness (whether a user could correctly decide whether a data table is useful), (2) efficiency (how fast a user could make the previous decision) and (3) user engagement when using the above summaries and standard, text-only, query-biased summaries.

We retrieved tables from the Australian Bureau of Statistics and showed these to three participants, naïve to our study; from these we elicited 60 queries, which we ran over the collection [4]. We selected 17 of these queries where at least two tables with query terms were retrieved in the top-10 results. Two of the selected queries were used as a warm-up task for which the data were not recorded, leaving fifteen queries in total for the experiment. For each query, we also wrote a short description of an information need.

To generate text-only query-biased summaries, we selected cells with query terms and the cells adjacent; concatenated the content of selected cells; highlighted query terms in the concatenated string;

	Text	Tabular/ graphical
Wasted downloads	21.7%	20.7%
Missed opportunities*	12.3%	7.7%
Accuracy***	42.0%	56.7%
Confidence**	60.0%	68.7%

Table 1: Effectiveness of query-biased summaries. χ^2 test: $^*p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.001$.

and truncated to 256 characters. Tabular and graphical summaries were generated using the algorithm above. Figure 2 shows examples of each summary type.

Participants were first given a description of the experiment and asked to stay focussed, read the summary enough to make an accurate judgement, and make decisions as quickly as possible.

This was followed by a series of tasks. Warm up tasks were clearly distinguished from the rest of the tasks. Each task consisted of an information need, a query, and query-biased summaries of two tables (either both text, or both tabular and graphical). The participant was first shown the information need and the query; then they were shown the two summaries, one after the other. Participants were asked to label each summarised table as "yes, useful"; "maybe, could be useful"; or "no, not useful". Task order was randomised, and summaries alternated between text-only and tabular and graphical.

After all tasks were completed, the tasks were presented again in the same order, but this time showing summaries of other kind.

At the end of the study was an optional short survey, in which we asked about familiarity with and frequency of use of search engines, and participants' opinions of the summaries they had seen. The familiarity and frequency questions were measured on a Likert scale of 1–5, with 1 being the least and 5 being the most familiar or frequent. The latter questions included:

1. Which type of summary is the most helpful to judge whether a dataset is worth downloading?
2. Which type of summary is the most helpful to find an answer for a specific question (e.g. Australia population 1999)?
3. Which type of summary is the most helpful to find an answer for a trend (e.g. population growth from 1950 to 2010)?
4. Which type of summary did you like the most?

Ethical review was by the CSIRO Social Science committee.

Results and evaluation. Ten people took part in the experiment, spending on average 20 minutes. The mean age of participants was 31 years (s.d. 13 years), six were male and four female. All participants were familiar with and frequent users of search engines with a mean of 4.4 and 4.8, respectively, on a 1–5 scale.

Recall that the participants answered "useful", "could be useful", or "not useful" for each query-biased summary. We count as a "wasted download" the occasions where a non-relevant set was labelled "useful" or "could be useful"—these are false positives—and count as a "missed opportunity" those occasions where a relevant set was labelled "not useful" (false negative). We also count occasions where the correct label was applied ("useful" for relevant; "not useful" for non-), and occasions when our participants were confident regardless of correctness ("useful" or "not useful", regardless of relevance). The results are shown in Table 1. We saw a significant increase in speed the second time participants saw a task, but no other effect due to task repetition.

On average users took slightly longer to judge a result with tabular and graphical summaries (13.2 s vs 10.8 s, $p < .001$), implying they spent more time reading these summaries than they did text.

The majority of participants preferred tabular and graphical summaries. Six of ten preferred them when deciding whether a data table is worth downloading; eight for finding an answer to a specific question; and all ten preferred them for finding trends. Seven of ten preferred them in general.

4. CONCLUSION

Despite the growth in online data portals, and the large amount of tabular data in personal and corporate systems, summaries of data rely on text and do not summarise the data itself. We have built a query-biased summarisation scheme for tabular data which either answers users' questions immediately, or supports their immediate decision to click through.

Our scheme uses simple type inference, identifies subsets of tables, and produces graphical summaries where possible. It does not make assumptions about semantics, and does not require any curatorial or authorial effort beyond generating a simple table with no extra metadata; this makes it appropriate in a variety of contexts including government, academic, or other aggregating portals.

Early results suggest these summaries help searchers identify relevant data (missed opportunities decrease) and distinguish relevant from irrelevant (accuracy increases). Participants in our studies were also more confident in their identification ("maybe" responses decreased even while accuracy increased). The tabular and graphical summaries were also popular: eight of ten users preferred them for factoid-type questions, and all preferred them for finding trends.

Future work includes extending table parsing and the type inference system, to handle less-regular tables (for example those with section headings, or unrecognised types). Other corner cases also need consideration—for example, graphing rows with identical dates. Finally, there is scope for a richer variety of plots, possibly including highlighting for outliers and trends. We also intend a larger-scale evaluation, with complete ranked lists, with more participants and over a large collection of tables.

References

[1] Sara Alspaugh, Marti A Hearst, Archana Ganapathi, and Randy Katz. Building blocks for exploratory data analysis tools. In *Proc. ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 9–17, 2013.

[2] Tereza Iofciu, Nick Craswell, and Milad Shokouhi. Evaluating the impact of snippet highlighting in search. In *Proc. SIGIR Workshop on Understanding the User*, 2009.

[3] Alicia Key, Bill Howe, Daniel Perry, and Cecilia Aragon. VizDeck: Self-organizing dashboards for visual analytics. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2012.

[4] Paul Thomas, Rollin Omari, and Tom Rowlands. Towards searching amongst tables. In *Proc. Australasian Document Computing Symposium*, pages 8:1–8:4, 2015.

[5] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, pages 2–10, 1998.

[6] Ryen W. White, Joemon M. Jose, and Ian Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Information Processing and Management*, 2003.