Tag Clouds for Software and Information Visualisation

Jessica Emerson Neville Churcher Andy Cockburn
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
jessica.emerson@pg.canterbury.ac.nz,{neville.churcher,andy.cockburn}@canterbury.ac.nz

ABSTRACT

We have extended the tag cloud metaphor to allow it to be applied to information and software visualisation. A number of issues, such as wide variation in tag length, have been addressed. We have developed a tool, TAGGLE, which implements our approach. In this paper, we present our visualisation technique and discuss the heuristic evaluation and report preliminary results from user trials employed to evaluate the approach and TAGGLE itself.

Keywords

Visualization, Visualization techniques, Tag Clouds, Information Visualization, Software Visualization

1. INTRODUCTION

A key distinguishing feature of information visualisation is that it involves quantities which do not have intrinsic geometric representations. A visualisation is based on computed geometry which, in turn, is based on one or more underlying metaphors. The effectiveness of an information visualisation technique will be affected by the metaphors that underpin it. Some general principles have become widely accepted. Shneiderman's information seeking mantra *Overview first, zoom and filter, then details-on-demand* is widely cited; *exploit human perception strengths* is embodied in techniques such as Chernoff facesand geons.

We are particularly interested in software visualisation — the application of information visualisation techniques to problems in the software engineering domain. This introduces a number of additional challenges — data is difficult to obtain, highly skewed, has large volumes and ranges, outliers, . . . — which we have discussed in previous work [8, 1].

In the next section, we describe the extension of tag clouds to the multivariate visualisation domain and introduce the tool, TAGGLE, we have implemented. Section 3 covers the evaluations we have conducted and our conclusions are presented in Section 4.

2. EXTENDING TAG CLOUDS

Tag clouds are often used to give an overview of the contents of speeches, documents and web sites. Wordle (`http://www.wordle.net`) and other tools have made it easy to create clouds but we believe their full potential for information visualisation has not yet been realised.

In "regular" tag clouds the only attribute which carries information is the font size of tags, typically reflecting the frequency with which the tag occurs in an underlying document or collection. Other elements (layout, colour, bounding shape, . . .) are generally only decorative. Our approach is to use properties such as these to allow the representation of more variables. In this way, a cloud becomes a representation of a multivariate data set, supporting users in gisting, searching, knowledge extraction and other tasks.

Tag clouds are particularly attractive in our software visualisation work because text labels are an intrinsic element. In other techniques, such as scatter plots or treemaps, text is problematic.

Our previous software visualisation work has been based on a pipeline approach to visualisation in which data undergoes a sequence of transformations in order to produce a visualisation presented to a user [7, 9, 1]. Some transformations involve operations on the data itself (extraction, selection, aggregation, . . .) which deliver the required data in a format appropriate to the visualisation metaphor and subsequent processing stages — we are not concerned with those further in this paper. Other transformations involve sets of mappings from quantities in the pipeline to attributes of the rendered visualisation. For example, a Java class may be represented by a red sphere whose radius is proportional to the number of public methods the class has. Our current work allows such a class to be represented by a tag (its name) whose colour denotes its age and whose size indicates the number of methods it has.

In the case of tag clouds, we can identify a number of such visual attributes of individual tags: text/label, location, size, colour (foreground/background), font (family, style, weight, . . .), transparency — even blink rate.

The cloud also has a number of attributes. These include layout — the order in which tags are placed and the algorithm used to determine individual tag locations — as well as the bounding dimensions and background colour.

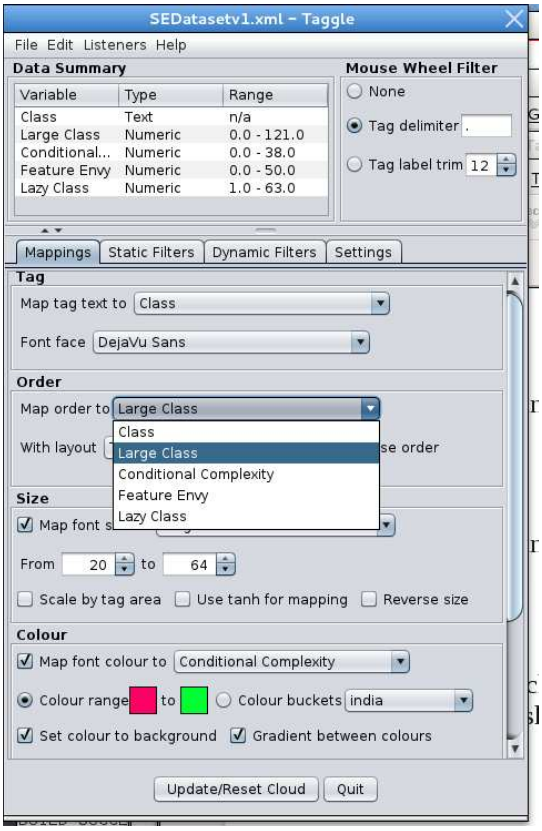


Figure 1: Taggle's control panel

Our approach, embodied in TAGGLE, involves allowing the user to explore data sets containing values for any number of variables. The user selects particular combinations of mappings from variables to visual attributes (e.g. $\{givenName \mapsto label, age \mapsto fontSize, weight \mapsto fontColour\}$) using TAGGLE's control panel as shown in Figure 1. These combinations may be saved and reloaded to enable the same analyses to be performed on multiple data sets.

A number of challenging issues arise when extending the concept to the visualisation domain and scaling up to data sets of realistic size. For example, tags with very long labels appear to be more prominent than they deserve to be. If there is considerable variation in the label length then this can be a significant factor.

As the number of tags increases, a point is reached where the display area is full. In order to fit in more tags one could simply reduce font sizes but this is ultimately limited by legibility concerns. Alternatively, labels can be "trimmed" to a maximum length with only the first (or last) parts displayed.

Taking advantage of known structure (e.g. the '`/`' in file names such as `/usr/local/bin` or the '`.`' in fully qualified names such as `java.lang.String`) provides a way to reduce label length at natural points.

Filtering (static) allows removing tags with a low Degree Of Interest (DOI) or replacing them with a minimal visual representation. Dynamic filtering allows tags matching a set of criteria to be dimmed or removed from a cloud.

TAGGLE provides features addressing these and other issues [4].

During a session with TAGGLE, users will typically:

- Create clouds (either from data files or by selecting tags from other clouds)
- Select appropriate mappings from variables in the data set to visual attributes.
- Choose layout algorithms (spiral, typewriter, ...) and layout order.
- Explore clouds: static and/or dynamic filtering; detailed inspection of all quantities associated with a tag (whether or not they are mapped to visual attributes); moving tags manually for detailed comparison; re-mapping to support evolving hypotheses.
- Save clouds and/or mapping sets for display (SVG) or further analysis (XML).

Examples, including videos of TAGGLE in action, are available at `http://www.cosc.canterbury.ac.nz/research/RG/svg/taggle/`.

3. EVALUATION

Our experiences with TAGGLE have satisfied us that there is indeed a rôle for tag clouds in software and information visualisation. However, we now need to identify tasks where tag clouds are particularly effective (and ineffective) in order to deploy them in the most productive contexts. Similarly, we wish to quantify baseline user performance factors and examine the usefulness of TAGGLE's features.

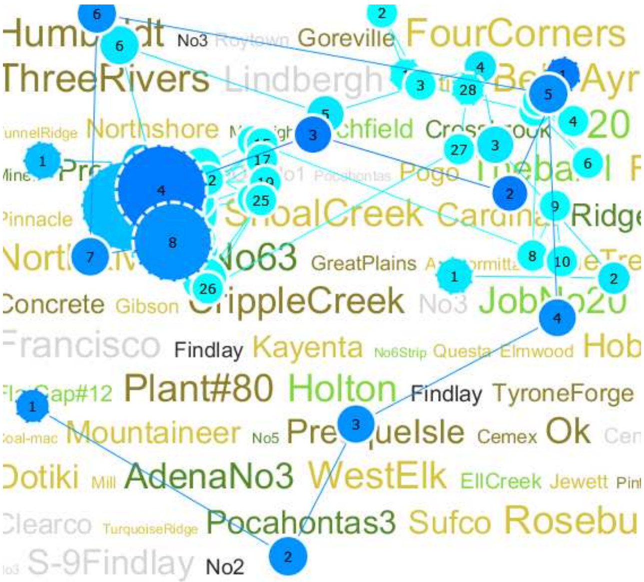
Evaluation in software engineering and visualisation contexts is notoriously challenging for a number of reasons. Typical task complexity and duration is higher than many "normal" interface interactions. Obtaining sufficient numbers of appropriately-skilled subjects is also problematic. Ideally, embedding a tool in a "real" user environment and logging usage would be possible but the impact of doing this with research tools is typically prohibitive.

Consequently, an exhaustive approach based on conventional hypothesis testing alone is impracticable at this stage. Later on, when results from our current studies are available, we expect to identify specific areas where this approach would be helpful.

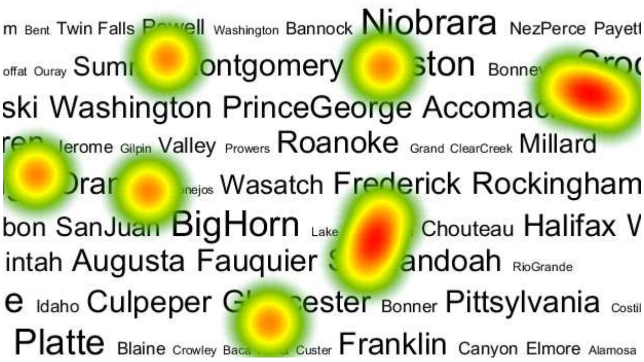
We have approached evaluation in three ways. A systematic mapping study was conducted to collate and review approaches to tag cloud visualisation and evaluation in the literature [3]. This informed the overall approach, including elements covering both whole tool & knowledge discovery and the visualisation technique itself, as well as the detailed experiment design.

We then conducted an heuristic evaluation [10, 11][1] of TAGGLE [2] using the ten heuristics proposed by Forsell and Jo-

[1] See also `http://www.nngroup.com`



(a) Gaze plot, multiple subjects



(b) Heat map, single subject

Figure 2: Sample subject tracking data

hansson [5]. These heuristics cover usability problems for information visualisation systems and are applicable to TAGGLE's tag cloud visualisations.

The evaluation identified a number of issues, the most significant of which were addressed before the user trials commenced. For example, the data summary table at the upper left of Figure 1 was added because the evaluators wanted a more accessible way to see the available variables and their ranges.

3.1 Trials

For our empirical studies we selected three activities strongly aligned with our visualisation research focus. Our subjects are second and third year computer science and software engineering students. Following initial training in tag clouds and TAGGLE, they performed a series of tasks. Subjects also completed the NASA TLX workload instrument. A Tobii eye tracker and video capture software were used to gather data for subsequent analysis.

3.1.1 Foreground/background colour

Our ability to perceive colour is influenced by a number of factors, the size of the coloured object being one of the main ones. Users of our visualisations will use tag colour as part of the gisting process (e.g. to gain an impression of the distribution of the variable mapped to colour) and other tasks. In correlations between size and colour, users need to be able to confirm statements such as "the smallest tags are the reddest."

Colours can be compared directly by dragging tags near each other or near the legend. Tag colour may be applied either to the text foreground or background bounding box. The canvas background colour may also be configured by the user.

A $2 \times 2 \times 2$ within-subjects experiment was conducted in which users were presented with combinations of colour placement (foreground, background), target tag size (small, large) and layout (spiral, typewriter) and asked to find a specific tag. Our expectation is that the larger area of colour in the background colour treatment will be associated with shorter completion times and that the difference will be greater for small tag sizes.

Preliminary results (from 5 subjects) appear consistent with our expectations: Tag selection time is faster using background colour ($\mu = 4.92s$) than foreground ($\mu = 6.38s$); tag selection time is faster for large ($\mu = 4.31s$) than small ($\mu = 6.98s$) tags; adding background colour improves selection time more for small ($\mu = 1.9s$) than large ($\mu = 1.0s$) tags. The effect of layout algorithm is currently less clear.

Figure 2(a) shows part of a cloud containing the target tag for a task (foreground tag colour, medium size, typewriter layout). The circles indicate points where a subject's gaze was focused and the numbers indicate the sequence in which the tags were studied. Results from several users (indicated by colour) are aggregated. Although users take differing times to reach the target, and travel by differing routes, the final gaze positions are co-located and have the longest dwell times.

3.1.2 Parallel mapping

Users are free to establish more mappings than are strictly necessary. A second experiment explores whether mapping both tag colour and tag size to a variable more effective than just mapping either one.

This is s $3 \times 2 \times 2$ within-subjects experiment. Users were presented with combinations of mappings (size, colour, size & colour), target tag size (small, large) and layout (spiral, typewriter) and asked to find a specific tag.

Anecdotal evidence, and our heuristic evaluation, suggests that the *parallel* mapping of both colour and size to a variable will be more effective. Figure 2(b) shows a heat map for a single user performing a task (size only, large tags, typewriter layout). The colours show the aggregated dwell time in each region. In this case, the subject has spent very little time looking at parts of the cloud other than the candidate targets meeting the search criteria but with different labels.

Preliminary results (from 5 participants) have the following

Figure 3: Knowledge discovery task in progress

mean selection times (seconds), suggesting that the parallel mappings improve performance for both layouts.

Mapping	Spiral	Typewriter
size only	4.8	4.5
colour only	4.3	4.6
size & colour	4.0	3.9

3.1.3 *Knowledge discovery*

Since TAGGLE is intended to be deployed (stand-alone or integrated with IDEs) as part of a software visualisation environment we included a knowledge discovery activity from the SE domain. Such activities are described in the literature[12, for example].

The data set used contains fully qualified class names from a Java program, together with the strengths of a number of code smells [6] — quantities which suggest software has flaws which require refactoring.

Subjects performed a number of benchmark tasks including: finding classes matching smell combination criteria; describing distributions of particular smells; detecting correlated smells and outliers;m discussing the similarities and differences between classes.

Figures 1 and 3 show a typical usage of TAGGLE in this activity. Figure 1 shows TAGGLE's control panel: the user is in the process of mapping the layout order to descending values of Large Class smell and tag colour has been mapped to *Conditional Complexity* smell. The corresponding cloud, shown in Figure 3 illustrates the correlation between these two smells and the skewed distribution of conditional complexity values.

4. CONCLUSIONS

We have extended the tag cloud model to support multivariate information and software visualisation. TAGGLE incorporates the extended model and allows user to design and explore data sets. It also includes features which address challenges arising from large or complex data sets. A heuristic evaluation has been conducted and we are currently completing user trials which include obtaining eye tracking data. Preliminary results indicate that our tag cloud model will prove a useful addition to the range of available techniques and we are encouraged to continue its development and to apply it in our software visualisation applications.

5. REFERENCES

[1] N. Churcher and W. Irwin. Informing the design of pipeline-based software visualisations. In S.-H. Hong, editor, *APVIS2005: Asia-Pacific Symposium on Information Visualisation*, volume 45 of *Conferences in Research and Practice in Information Technology*, pages 59–68, Sydney, Australia, Jan. 2005. ACS.

[2] J. Emerson. Tag clouds in software visualisation. MSc thesis, Department of Computer Science and Software Engineering, University of Canterbury, 2014. in progress.

[3] J. Emerson and N. Churcher. Evaluation of tag cloud visualisation: A systematic mapping study. Technical Report TR-COSC 01/13, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand, 2013.

[4] J. Emerson, N. Churcher, and C. Deaker. From toy to tool: Extending tag clouds for software and information visualisation. In J. Dietrich and J. Noble, editors, *22nd Australian Conference on Software Engineering*, pages 155–164, 4–7 June 2013.

[5] C. Forsell and J. Johansson. An heuristic set for evaluation in information visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 199–206, New York, NY, USA, 2010. ACM.

[6] M. Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.

[7] W. Irwin and N. Churcher. XML in the visualisation pipeline. In D. D. Feng, J. Jin, P. Eades, and H. Yan, editors, *Visualisation 2001*, volume 11 of *Conferences in Research and Practice in Information Technology*, pages 59–68, Sydney, Australia, Apr. 2002. ACS. Selected papers from 2001 Pan-Sydney Workshop on Visual Information Processing.

[8] W. Irwin and N. Churcher. Object oriented metrics: Precision tools and configurable visualisations. In *METRICS2003: 9th IEEE Symposium on Software Metrics*, pages 112–123, 2003.

[9] W. Irwin and N. Churcher. Object oriented metrics: Precision tools and configurable visualisations. In *METRICS2003: 9th IEEE Symposium on Software Metrics*, pages 112–123, Sydney, Australia, Sept. 2003. IEEE Press.

[10] J. Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 373–380. ACM Press, 1992.

[11] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 206–213. ACM Press, 1993.

[12] C. North, P. Saraiya, and K. Duca. A comparison of benchmark task and insight evaluation methods for information visualization. *Information Visualization*, 10(3):162–181, 2011.