Toward Building Aesthetic, useful and Readable Tag Clouds for Websites

Jakub Marszałkowski, Łukasz Rusiecki, Maciej Drozdowski and Hubert Narożny

Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland

Keywords: Web Page Optimization, Tag Clouds, 2D Packing, Data Visualization, Usability.

Abstract: Tag clouds provide a graphical method of summarizing content of a text document (e.g. of a web page) with a set of phrases projected onto a plane. In this paper we consider building aesthetic tag clouds algorithmically for website use. General design choices in tag-cloud construction are analyzed. State of the art will be outlined along the lines of these design choices. Special requirements imposed on tag clouds used on web sites are presented. Rules of beautiful page setting existing in typography are discussed and subsequently applied in an attempt to quantify aesthetic aspect of tag cloud appearance. The quantification is performed with the goal of constructing an objective function to be optimized in tag cloud construction. Then, a mathematical formulation for tag clouds construction is given. Finally, algorithms constructing tag clouds by optimization are given.

1 INTRODUCTION

Internet-related problems instigate research in many areas. One of them is combinatorial optimization. As examples, consider e-business problems of Internet shopping optimization (Błażewicz and Musiał, 2011), website layout optimization for the purpose of flexible placement of the future advertisements (Marszałkowski and Drozdowski, 2013). Yet, probably the most studied area of the research in this field is Internet advertising scheduling and advertising display positioning with papers from as old as (Aggarwal et al., 1998) to the recent ones like (Ahmed and Kwon, 2014). Tag cloud construction is also a matter of solving a combinatorial optimization problem, however, the objectives are very different than in the classic combinatorial problems.

Basically, tags are phrases representing or summarizing content of a text document such as a web page. Tag cloud is a graphical depicting of the tags as just a set of words/phrases projected onto a plane. An example tag cloud from Amazon website is shown in Figure 1. Tags and tag clouds originated from social websites, but they gained already a wide usage over the entire Internet. More details on tag cloud usage and history can by found in (Viégas and Wattenberg, 2008). Tag cloud construction also receives growing interest of the researcher community as shown in the next section.

In this paper we analyze the problem of constructing tag clouds for use on web pages that are visually acceptable or hopefully even pleasing. A first step in tag cloud creation is preparation of tags themselves: selection, grouping, clustering, weighting, etc. Here it is assumed that the set of tags is given and their rendering in two dimensions is studied. Methods of digesting the text and extracting the tags rest in text mining area and are beyond the scope of this paper. The problem of rendering the tags into a tag cloud is formulated as a combinatorial problem with specific objectives and constraints. Further organization of this text is the following. In Section 2 tag clouds in general and ones for websites are discussed. Design options and the choices taken in the past are surveyed. Requirements for tag clouds for web usage and client side generation are discussed. Section 3 provides a mathematical formulation of a tag cloud construction problem, as well as, the first approach to algorithms solving the problem.

2 TAG CLOUDS

In this section we present results of the research on tag cloud formation and usability. Then we discuss requirements for website tag clouds. Finally, the requirements of and status quo in web browsers flexibility are studied.

2.1 State of the Art

In tag cloud construction there are several design choices determining appearance and usability of tag clouds. In particular these are:

Marszałkowski J., Rusiecki Ł., Drozdowski M. and Narożny H..
Toward Building Aesthetic, Useful and Readable Tag Clouds for Websites.
DOI: 10.5220/0005116302300235
In *Proceedings of the 11th International Conference on e-Business* (ICE-B-2014), pages 230-235
ISBN: 978-989-758-043-7

Table 1: Summary of packing choices in tag clouds (See Section 2 for details).

Cloud	1. Tag ordering	2. Cloud shape	3. Tag shape	4. Tag rotation	5. Vertical alignment
Amazon	alphabetical	rectangle	rectangle	none	baseline
Kaser	packing	rectangle	rectangle	none	limited
Kuo	alphabetical	rectangle	rectangle	none	baseline
Fujimura	context	irregular	rectangle	none	background
Seifert	packing	given polygon	rectangle	none	free
Wordle	opt. alphabetical	irregular	font body	configurable	free
Cui	context	irregular	rectangle	none	free
Nguyen	alphabetical	given borders	rectangle	none	background
This paper	packing	rectangle	rectangle	none	baseline

1. How tags are sorted. Identified choices are: alphabetically, by importance, by context, packing-decided. The latter means that tags may be re-ordered for better packing quality.
2. Shape of the entire cloud area. Possible options: rectangular, irregular, other (given polygons, defined borders).
3. What kind of figure the tags are. Options: rectangular boxes, or character body. The former means that bounding boxes of the tags rendered in some given font are used. The latter means using the irregular shapes of characters in the given font. This allows for advanced tag alignment in free spaces of the letter bodies.
4. Tag rotation: disallowed or allowed.
5. Vertical tag alignment. Identified options are: sticking to the typographical baselines, limited by the algorithm properties, free - leading to 2D packing, forced by tag cloud background.

Results of the outermost design decisions can be compared in Figures 1 and 2 (cf. the design decisions outlined in Table 1). Let us note that the above choices resemble variants in combinatorial problems of packing and cutting. It will be discussed in the following text. There are also further design-choices possible. For example, it can be related to the use of colors or fonts (typefaces, sizes, weights and styles). We assume that fonts are given as input, coming from the tag preparation step. Note that use of colors to distinguish tags may be a bad idea for users with color-impaired sight. Hence, we assume that tags are essentially monochromatic (e.g. black) on a contrasting (e.g. white) background.

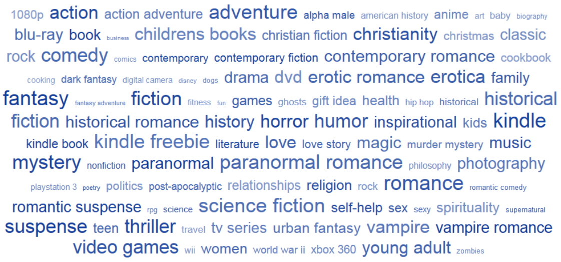
1080p action action adventure adventure alpha male american history anime art baby biography
blu-ray book business childrens books christian fiction christianity christmas classic
rock comedy comics contemporary contemporary fiction contemporary romance cookbook
cooking dark fantasy digital camera disney dogs drama dvd erotic romance erotica family
fantasy fantasy adventure fiction fitness fun games ghosts gift idea health hip hop historical historical
fiction historical romance history horror humor inspirational kids kindle
kindle book kindle freebie literature love love story magic murder mystery music
mystery nonfiction paranormal paranormal romance philosophy photography
playstation 3 poetry politics post-apocalyptic relationships religion rock romance romantic comedy
romantic suspense rpg science science fiction self-help sex sexy spirituality supernatural
suspense teen thriller travel tv series urban fantasy vampire vampire romance
video games wii women world war ii xbox 360 young adult zombies

Figure 1: Tag cloud amazon.com/tags.

Tag clouds construction attract increasing interest of researchers. (Kaser and Lemire, 2007) submitted an idea of nested HTML tables in order to build tag clouds well using given rectangular space. (Kuo et al., 2007) presented application of simple tag cloud to summarize results of a query over a database. (Fujimura et al., 2008) proposed use of a topographical map as background to visualize large scale tag clouds (5000 tags, 10000x10000 pixels). Position of tags is determined by the map, height on the map reflects tags importance. (Seifert et al., 2008) worked on fitting tags clouds into polygons, and proposed four algorithms. Out of these four algorithms they chose ones for best usability and aesthetic parameters. Wordle (Viegas et al., 2009) is probably the best-known web-based tool for data visualizations in the form of a tag cloud. It allows to set several parameters like rotation or sorting, but always justifies tags to shapes of characters and outputs irregular tag clouds. (Cui et al., 2010) proposed tag clouds preserving context using color for visualizing trends. (Nguyen and Schumann, 2010) explored putting tags into shapes resembling maps to achieve geo-tagged data exploration. The choices made by the authors of the above papers according to terminology introduced at the beginning of this section are summarized in the Table 1. Out of these papers only the first two consider designing tag clouds for website use.

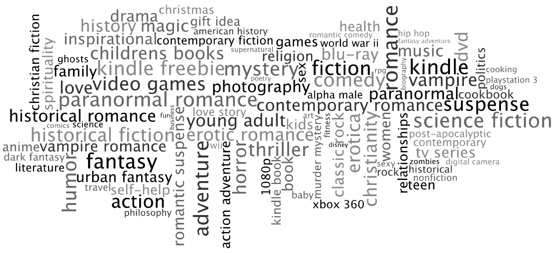

Figure 2: Amazon tags rendered into a cloud by Wordle.

A few studies verifying effectiveness of tag clouds and user experience have been conducted. List of tasks that tag clouds can support: Search, Browsing, Impression Formation and Recognition/Matching is given by (Rivadeneira et al., 2007). Out of these, the last one means verifying whether tag cloud is representing particular subject. Note that only Search is goal-oriented task, while the remaining are rather free browsing tasks. (Halvey and Keane, 2007) performed simple experiment with time necessary for finding certain tag, and they found that alphabetical list is actually faster. They also conclude that users rather scan than read tag clouds. When testing clouds obtained from their algorithm, (Seifert et al., 2008) used a different approach. Namely, they asked users to point three most important tags and measured the correctness. Though this seems a better idea for evaluating tag clouds, their experiments were strongly related to their algorithms and give no general insights. (Rivadeneira et al., 2007) on the basis of their results, conclude that font size and location affect low-level memory processes, while layout high-level ones, such as impression formation. They suggest to focus on the layout of tag cloud. Research of (Bateman et al., 2008) did not tackle the layout matters, instead font related parameters were tested leading to conclusions that larger and stronger fonts draw more users attention, while color although being well recognized proves difficulties in visualizing importance. (Lohmann et al., 2009) performed several experiments on performance of certain tasks involving various cloud layouts. They confirm earlier findings of (Halvey and Keane, 2007) that finding a specific tag is fastest with alphabetical sorting and that users are scanning rather than reading. Yet, their other experiments show that for finding most important tags, recalling tags, etc. layout plays important role.

The above presented research was focused on goal-orientated tasks, which are easier to measure, as opposed to free browsing tasks. However, browsing is an important application of tag clouds.

2.2 For the Web

In authors' opinion, tag clouds for websites have to meet additional requirements. Website space is always rectangular and scarce so it should be used wisely. This gives a preference to tag clouds filling a rectangular envelope well. As websites usually use column layout (Marszałkowski and Drozdowski, 2013) horizontal size of a tag cloud is fixed, while the vertical size can be changed, thus moving the component below up or down a little. This characteristic resembles strip packing problems.

A tag cloud for a website should use standard technologies, making a reasonable trade-off between fancy looks and the simplicity of the code. This has two reasons: Firstly, it is a matter of the ease of implementation. Secondly, not only humans read websites and making website content available to the robots is of great importance (see (Marszalkowski et al., 2014)). Using HTML with JavaScript (JS) and CSS as simple as possible seems to be a natural decision here. This simplifies some of the further choices: Though the use of exact tag shapes or tag rotation are possible in most modern browsers, they are not standard and cannot be guaranteed to work perfectly the same way for every client. Hence, they should be discouraged. The same argument can be applied in prefering the alignment to the baselines over the freedom of arbitrary 2D packing. Tags on a baseline will be considered just as line of text by the robots. Taking into account the results of the studies demonstrating that users scan lines of the clouds (see Section 2.1), the use of baselines will make reading tags more efficient.

Next we come to the choice of tag ordering. It was already mentioned that alphabetical clouds perform worse in the speed of searching compared to lists, so why to use them? Moreover, alphabetical ordering significantly restricts flexibility of packing in two following ways. Firstly, since tags cannot be reordered the only remaining option is to choose where to put a line break. Secondly, for the same reason use of differing font sizes must remain limited, as tags of the smallest font cannot be moved away from the lines made very tall by the tags of the greatest font size. To achieve any reasonable visual quality tags have to be rearranged, i.e. the sequence tags should follow packing.

Our design recommendations are: 1) tags are reordered with packing, 2) minimum waste of the rectangular area is desired, 3) tags are rectangular boxes, 4) rotation is not allowed, 5) tags fit between baselines (shelves). Although it may seem that in most cases simplifying choices were made, we end up with a problem that can be expected to rest in NP-hard class. Thus, it can be expected that optimum solutions (e.g. in the sense of used area) can be delivered by exponential-time algorithms. The current recommendation encompasses bin packing problems or strip packing which can be solved by use of shelf algorithms, or metaheurisitcs (Burke et al., 2006).

2.3 Client Side

In times of more and more personalized content each user can get a different set of tags. But there is more than that to significantly affect packing of the tags. Namely, clients may have different dimensions (in pixels) of the same tag depending on the browser, system and fonts installed. We conducted an experiment into browser font rendering dispersion to verify our intuitions. We tailored 6 benchmark tags testing different methods of defining look of text with CSS: fonts, font stacks, size and weight. A script measuring tag sizes was installed on a production web site and in course of two days responses from 4201 different clients were registered.

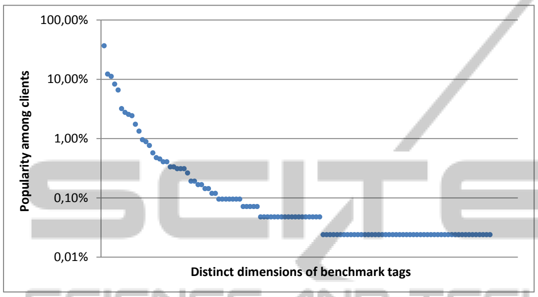

Figure 3: Distribution of dimensions of tags over the measured Internet users.

In the gathered data we identified 112 distinct sizes for the benchmark tags. The results are shown in the Figure 3. As could be expected we found that the distribution of tag sizes follow the power law (with fit quality $R^2 = 0.9774$). Most of users use browser/systems combinations that render the tags in less than a dozen of the most popular sizes. On the one end, the three most popular font sizes are found on, respectively, 36.61%, 12.21% and 11.19% of clients. On the opposite end, sizes with less than 1% popularity form a long tail of 101 different values. More so, tag sizes on mobile devices differ much more than on standard desktop/laptop computers (even two-three times). The result lead to a conclusion that we have to adjust tag cloud construction to the tag sizes measured at client side.

Algorithmic building of tag clouds can be moved to client side by meeting just a few requirements. The implementation has to use JavaScript (JS). Although other choices are possible, only JS has sufficient market penetration. Moreover, JS works on the elements in DOM structure preserving readability of the tag cloud for the robots. A disadvantage is that the algorithm constructing a tag cloud must run in very limited time, i.e. in th order of tenths of a second. There is plenty of research showing that users do not want to wait for downloading web-page content from the Internet and rendering it, because they quickly lose interest. An up-to-date survey is given in (Marszalkowski et al., 2014). Since the performance of the client browser is unknown, the algorithm must stop when time limit is reached, and give a valid solution.

3 2D PACKING FOR BETTER QUALITY

In this section we formulate tag cloud construction problem as an optimization problem. What is novel in this approach is a proposition of reaching back to the rules of typography, which tell how to typeset readable text objects looking good aesthetically. One of such rules, already introduced for other reasons, is the use of baselines. Other rule is a desire for good tonal weight (also known as the typographic color). This means even distribution of the mass of gray in case of black letters on white background (Bringhurst, 1996; Eckersley et al., 2008). A typographer usually has to squint to asses that. An advantage for building tag clouds is that this black color dispersion is measurable and can be included in a mathematical model, which is rare for the rules of beauty. The tonal weight can be measured in HTML by reading colors of a pixel in the canvas element. A canvas would be impractical for building whole tag clouds, and would contradict the requirements declared earlier. Luckily, a tag can be put in canvas, have its tonal weight measured and recorded. Tonal weights of entire tag cloud, or its sections, can be calculated from tonal weights of single tags and their dimensions.

3.1 Mathematical Model

Tag cloud construction problem can be formalized as a Nonlinear Programming (NLP) optimization problem as follows. Given is width of a tag cloud W and a set of tags $T = \{t_1, \ldots, t_n\}$. Dimensions of a tag i are x_i and y_i. Tonal weight of each tag can be measured as a sum of tonal weights of its pixels:

$$a_i = \sum_{1 \le x \le x_i, 1 \le y \le y_i} b[x,y] \qquad (1)$$

where $b[x,y]$ is the tonal weight of the pixel at a position x,y, calculated as:

$$b = 1 - \frac{R+G+B}{3*255} \qquad (2)$$

R, G and B are values of color bytes for the pixel.

Let f_i represent the number of the shelf (baseline) where tag t_i was placed, and let m be the total number of shelves. Let Z_j represent a set of tags placed on a shelf j, i.e.: $Z_j = \{t_i : f_i = j\}$. Tonal weight of shelf j can be calculated from tags on it and its height:

$$\alpha_j = \frac{\sum_{t_i \in Z_j} a_i}{h_j * W} \qquad (3)$$

where in denominator we have dimensions of the shelf: height $h_j = \max(x_i | t_i \in Z_j)$, and width W. This causes that the free space will be reflected in the value of α. For example, if tags have large differences in height, or shelf will be under-utilized, and large empty areas shall result in low value of α_j. With the scores α_j of shelves j we construct objective function quantifying the differences in tonal weights between shelves:

$$\min \sum_{j=1}^{m} \left(1 - \frac{\alpha_j}{(h_j * W)}\right)^2 \qquad (4)$$

In (4) a deviation from the maximal possible tonal weight is calculated. It follows implicitly from (4) that if (4) is minimized then also the height of tag cloud will be minimized. Finally, it is required that all tags assigned to a some shelves fit the shelves:

$$\sum_{t_i \in Z_j} y_i < W \qquad \forall\, j = 1, \ldots, m \qquad (5)$$

Note that in this approach neither shelves ordering nor tags ordering on shelves matter. These can be rearranged after the packing for example to move more important tags to areas more frequently scanned by users.

3.2 Algorithms for Tag Cloud Optimization

To solve this problem to optimality Branch and Bound (B&B) algorithm was developed (Lawler and Wood, 1966). Since our problem involves an additional criterion of minimizing the number of shelves, the B&B algorithm was first calculating the minimum number of shelves and only then was it minimizing the tonal weight inequalities with objective function (4). Obviously B&B exponential running time renders it not usable in practice. It was solving instances of up to 20 tags, in execution times up to 7 minutes. However, the B&B algorithm is necessary to allow measuring optimality gap of other algorithms.

Several well known low-level heuristics greedy like FirstFit (FF), BestFit (BF), WorstFit (WF) (Burke et al., 2006) were used. Also modified version of the FF algorithm First Fit Greedy Two (FFG2) was developed. FFG2 follows a simple idea: when placing an element of size x on a shelf, and the space left will be less than the narrowest remaining element, look for two elements of sizes nearest to $\frac{x}{2}$, and check if the pair fits better (leaving less waste) than the element of size x. With the use of earlier sorting of the elements by width, the checking for these two elements can be achieved in constant time, and the algorithm has the overall complexity $O(n \log n)$ of FF algorithm. For all four algorithms their versions using different ordering of elements were used: decreasing height (DH), decreasing width (DW), decreasing tonal weight (DT), making it twelve algorithms in total.

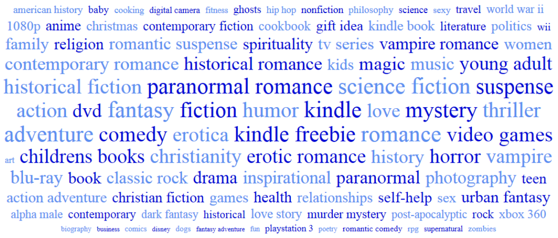
american history baby cooking digital camera fitness ghosts hip hop nonfiction philosophy science sexy travel world war ii
1080p anime christmas contemporary fiction cookbook gift idea kindle book literature politics wii
family religion romantic suspense spirituality tv series vampire romance women
contemporary romance historical romance kids magic music young adult
historical fiction paranormal romance science fiction suspense
action dvd fantasy fiction humor kindle love mystery thriller
adventure comedy erotica kindle freebie romance video games
art childrens books christianity erotic romance history horror vampire
blu-ray book classic rock drama inspirational paranormal photography teen
action adventure christian fiction games health relationships self-help sex urban fantasy
alpha male contemporary dark fantasy historical love story murder mystery post-apocalyptic rock xbox 360
biography business comics disney dogs fantasy adventure fun playstation 3 poetry romantic comedy rpg supernatural zombies

Figure 4: Example initial results built of tags from Amazon.

The algorithms were tested on eight sets of tags taken from real world websites. The best performing algorithms were WF (DH and DW) and FFG2 (DT and DH) with a gap from 0.02% to 18.71% to the result of B&B. This is not really surprising as WF algorithm is making choices equalizing utilization of shelves, and FFG2 should on average pack slightly better than FF an BF algorithms. An example of their results, a cloud built from the same set of tags from Amazon is presented in Figure 4. Execution times were of order of microseconds, which means that these algorithms can be run many times as part as some hiper- or metaheuristic algorithm before becoming noticeable for the user. The highest observed gaps of over 15% encourage building such algorithm.

4 CONCLUSIONS AND FUTURE WORK

In this paper the idea of constructing aesthetic tag clouds for websites by algorithmically following rules of typography was presented and justified. An important contribution is inclusion of the typographical rule of good tonal weight in the objective function. This metric of quality of text object is easily measurable, and thus can be included in the mathematical model and optimized. Such a model was given and work on algorithms can be performed in future on its basis.

Our experiment showing diversity of sizes for the same tags on different client caught quite few mobile devices, and perhaps another experiment for more detailed insight in this area should be performed as a future work. Also some choices of the parameters of tag clouds, like for example lack of rotation of the tags, that was not covered in research to this date, could have quality of user experience experimentally verified. Algorithms proposed here were very quick which opens chances for algorithms for constructing even better tag clouds in longer time.

ACKNOWLEDGEMENTS

The research was partially supported by the NCBiR PolLux grant (IShOP project).

REFERENCES

Aggarwal, C. C., Wolf, J. L., and Philip, S. Y. (1998). A framework for the optimizing of www advertising. In *Trends in Distributed Systems for Electronic Commerce*, pages 1–10. Springer.

Ahmed, M. T. and Kwon, C. (2014). Optimal contract-sizing in online display advertising for publishers with regret considerations. *Omega*, 42(1):201–212.

Bateman, S., Gutwin, C., and Nacenta, M. (2008). Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 193–202. ACM.

Błażewicz, J. and Musiał, J. (2011). E-commerce evaluation–multi-item internet shopping. optimization and heuristic algorithms. In *Operations Research Proceedings 2010*, pages 149–154. Springer.

Bringhurst, R. (1996). *The elements of typographic style*. CRC Studio.

Burke, E. K., Hyde, M. R., and Kendall, G. (2006). Evolving bin packing heuristics with genetic programming. In *Parallel Problem Solving from Nature-PPSN IX*, pages 860–869. Springer.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., and Qu, H. (2010). Context preserving dynamic word cloud visualization. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 121–128. IEEE.

Eckersley, R., Angstadt, R., Ellertson, C. M., and Hendel, R. (2008). *Glossary of typesetting terms*. University of Chicago Press.

Fujimura, K., Fujimura, S., Matsubayashi, T., Yamada, T., and Okuda, H. (2008). Topigraphy: visualization for large-scale tag clouds. In *Proceedings of the 17th international conference on World Wide Web*, pages 1087–1088. ACM.

Halvey, M. J. and Keane, M. T. (2007). An assessment of tag presentation techniques. In *Proceedings of the 16th international conference on World Wide Web*, pages 1313–1314. ACM.

Kaser, O. and Lemire, D. (2007). Tag-cloud drawing: Algorithms for cloud visualization. *arXiv preprint cs/0703109*.

Kuo, B. Y., Hentrich, T., Good, B. M., and Wilkinson, M. D. (2007). Tag clouds for summarizing web search results. In *Proceedings of the 16th international conference on World Wide Web*, pages 1203–1204. ACM.

Lawler, E. L. and Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719.

Lohmann, S., Ziegler, J., and Tetzlaff, L. (2009). Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Human-Computer Interaction–INTERACT 2009*, pages 392–404. Springer.

Marszałkowski, J. and Drozdowski, M. (2013). Optimization of column width in website layout for advertisement fit. *European Journal of Operational Research*, 226(3):592–601.

Marszalkowski, J., Marszalkowski, J. M., and Drozdowski, M. (2014). Empirical study of load time factor in search engine ranking. *Journal of Web Engineering*, 13(1-2):114–128.

Nguyen, D.-Q. and Schumann, H. (2010). Taggram: Exploring geo-data on maps through a tag cloud-based visualization. In *Information Visualisation (IV), 2010 14th International Conference*, pages 322–328. IEEE.

Rivadeneira, A. W., Gruen, D. M., Muller, M. J., and Millen, D. R. (2007). Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998. ACM.

Seifert, C., Kump, B., Kienreich, W., Granitzer, G., and Granitzer, M. (2008). On the beauty and usability of tag clouds. In *Information Visualisation, 2008. IV'08. 12th International Conference*, pages 17–25. IEEE.

Viégas, F. B. and Wattenberg, M. (2008). Timelines tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52.

Viegas, F. B., Wattenberg, M., and Feinberg, J. (2009). Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1137–1144.