2014 18th International Conference on Information Visualisation

RadCloud: Visualizing Multiple Texts with Merged Word Clouds

Michael Burch[1], Steffen Lohmann[1], Fabian Beck[1],
Nils Rodriguez[1], Lorenzo Di Silvestro[2], Daniel Weiskopf[1]

[1]*University of Stuttgart, Germany*
{*michael.burch,fabian.beck,daniel.weiskopf*}*@visus.uni-stuttgart.de,*
steffen.lohmann@vis.uni-stuttgart.de

[2]*University of Catania, Italy*
disilvestro@dmi.unict.it

***Abstract*—Word clouds are a popular means for summarizing text documents. They usually visualize the word frequencies from single text sources, sometimes along with other attributes. However, the visualization of several text documents in one word cloud has rarely been addressed so far. This paper presents RadCloud, a technique for text visualization based on multiple word clouds merged into a single view. Inspired by the RadViz approach, the words are radially arranged in an overlap-free layout. The text sources are indicated by the spatial word arrangement and stacked bar charts. The approach has been implemented in an interactive text visualization tool and its usefulness is illustrated by an example.**

***Keywords*-Text visualization, word cloud, tag cloud, information visualization, radviz.**

I. Introduction

A word cloud (also known as tag cloud) is a popular technique for visualizing texts. It is basically a weighted list of words with a certain spatial arrangement (sequential, circular, clustered, etc.). The font sizes of the words indicate their importance or occurrence frequency in the underlying text data. Sometimes, further visual properties are varied, such as the font color or word orientation.

A common application area of word clouds is text summarization, where they are typically used to depict the words from selected texts. They often serve as a starting point for a deeper analysis and help to judge whether a text is relevant to a specific information need [12], [25]. However, current word cloud implementations provide limited support to visualize several texts at once and to compare their words and word frequencies.

In this paper, we present RadCloud, an approach to overcome this limitation by visualizing words from different text sources in a single merged word cloud. Inspired by the design of the RadViz technique [13], the words are displayed inside a circle, with the text sources serving as anchor points on the circle circumference. The relevance of the words per source is indicated by their position in the circle and additionally with stacked bar charts that complement the visualization.

II. Related Work

Several improvements and extensions to word clouds have been proposed in the last couple of years. Many of them address layout issues. For instance, Kaser and Lemire [14] present methods to reduce and balance the white space in HTML-based word clouds. Seifert et al. [21] also propose space-filling word cloud algorithms, but use a different layout strategy to cope with convex polygons as boundaries. Yet another layout strategy is applied in the popular word cloud generator Wordle [7], which has been adapted with some modifications in the works on ManyWordle [15] and Rolled-out Wordles [23].

Another line of research investigates the application of clustering techniques in word clouds to indicate word relatedness by spatial distance. Straightforward approaches arrange the clustered words line-by-line [11], [20] or in a force-directed layout [4]. Other works apply multidimensional scaling to reflect the semantic relatedness of words [19], [26], or use topographical word landscapes similar to the ThemeScape visualization [9].

There are also attempts to explicitly depict relationships in words clouds, either by adding links to related words [22] or by using interactive highlighting [12], [17]. Tree Clouds [10] combine word clouds with trees to connect the words, whereas Prefix Tag Clouds [2] make use of prefix trees to group different word forms.

While these works cluster or group the words based on their co-occurrence or some other measure, they do not distinguish between different text sources. Usually, the words are treated as if they would all come from the same text document. This is also true for approaches that add a temporal dimension to word clouds, for instance, by using sparklines [16] or histograms [17] in order to indicate changes in the word frequencies over time. While these visualizations illustrate the evolution of words in different text documents, the text documents themselves are not distinguished in the word clouds. The same limitation holds for the work of Cui et al. [6] who coupled a trend chart with word clouds to illustrate the temporal evolution of words.

1550-6037/14 $31.00 © 2014 IEEE
DOI 10.1109/IV.2014.72

One possible way of presenting words along with their text source information is to use small multiplies of word clouds, each for one text document. For instance, OpinionSeer [27] uses this approach to visually summarize and compare the reviews of hotel customers. A similar approach can be found in the text analysis system POSvis [25], where each small multiple is dedicated to a specific part-of-speech category. However, the same words are displayed several times in these solutions, and it is rather difficult to compare individual words and their relevances for the different texts.

A similar limitation is given in Parallel Tag Clouds [5] that combine the ideas of word clouds, small multiples, and parallel coordinates by making each coordinate a weighted list of words. While the row-based presentation and visual linking are very useful, words still appear multiple times and interaction is required to identify and understand their relevance with regard to the different text sources.

An exception in this regard is the word cloud visualization of the ManyEyes website [24] that is capable of showing words from different texts in one word cloud, simply indicating the text source of each word by its font color. However, apart from the fact that words appear redundantly in the cloud, the approach does not scale well if more than two texts are visually compared.

III. Text Processing

In contrast to related work, we aim at a solution that visualizes the words from several text documents in a single word cloud. Each word should only appear once, while the text sources of the words should be visually indicated.

Before we can generate such a visualization, the text sources must first be processed. The text processing consists of the following (partly optional) steps in our approach: tokenization, named entity recognition, word extraction, and relevance calculation.

Our user interface consists of a series of dialogs that lead the users in the text processing and let them control and configure the parameters of each step in accordance with the analysis context and goals.

A. Input Data

The data consists of a set of text documents that should be visualized with the RadCloud approach. In the simplest case, the text documents are not further organized. In more advanced cases, the documents may be grouped into different categories, which would be reflected by the directory structure in our approach: Each category is represented by one directory containing the text documents assigned to it, with the category name being the directory name.

The texts are tokenized, i.e. they are split at whitespace characters and punctuation marks to get the individual words. The list of tokens serves as the input for the text processing steps.

B. Named Entity Recognition

We included the option to run a named entity recognizer (NER) on the texts. The NER adds annotations to the words that can improve the word extraction (see below). We use a .NET implementation of the Stanford NER [8] for this purpose. However, this processing step is optional, as it can be time-consuming.

If users want to perform a named entity recognition, they have to select a method and trained model. By default, our implementation offers the methods and models deployed with the Stanford NER library, in particular a model that is capable of identifying locations, persons, and organizations in English texts and that is trained on both the CoNLL and MUC data sets [8].

The text files are processed one-by-one by the NER component and saved along with the annotations in the target directory. Keeping all this information in memory would unnecessarily strain or even exceed the available resources. Furthermore, the disk storage ensures that the text annotations remain available for future analyses.

C. Word Extraction

In the word extraction step, users can decide if all words are considered or only those that were annotated as named entities in the previous step. In addition, they can determine if multiword expressions (e.g., “United Kingdom”) are treated as one word or as separate words and whether numbers should count as words.

As common in the generation of word clouds, we also use stop word lists to filter out irrelevant words, i.e., words like ‘the’, ‘is’, ‘at’ that usually do not carry meaning. We offer stop word lists for different languages, as provided by the CLEF project [1]. The lists are maintained in separate text files and can thus be easily edited or replaced by other stop word lists.

D. Relevance Calculation

Common word cloud generators use the words’ frequencies as indicator for their relevance. This is based on the simplified assumption that the most relevant words are those that occur most often in the text. While this assumption is acceptable for single text documents, it is no longer sufficient when generating word clouds for multiple text documents. Assuming there are two documents A and B, and A is much longer than B. If an unimportant word appears in every second paragraph of document A, it would get a much higher relevance than an important word appearing in every second paragraph of document B.

A straightforward way to tackle this problem is to multiply the term frequency (tf) with the inverse document frequency (idf). This takes into account in how many documents a word appears and changes its weight accordingly, i.e., the more documents it is contained in, the less relevant it is for a particular document. Next to the word frequency, we

therefore offer the tf-idf as an alternative metric in our approach.

We additionally save the index positions of the words in the text files, so that they can later be accessed in their context by the user. We use a dictionary to store this information, with the documents' relative file paths as keys. The values consist of the sorted lists of indices pointing to the positions in which the words occur in the texts.

In addition, we implemented a third relevance metric that can either be based on the tf or tf-idf. It uses a support vector machine (SVM) to determine the relevance of the words. In this approach, a vector is created for each document using a .NET implementation of libsvm [3]. Depending on the user's choice, either the tf or tf-idf values of the words are used in the SVM. In addition, a grid search is offered by the SVM implementation. It tests various settings around a start configuration and automatically determines the values that should lead to satisfying results. The detailed description of the SVM approach is, however, outside the scope of this paper.

Finally, the extracted words are saved along with all metadata and settings information in either a CSV or XML file. This allows us to reuse and adapt the data for later analyses or to utilize it in other tools. The file is also used for the generation of the RadCloud, which is described in the next section. The user can set a maximum count of words that should be saved from each category to reduce the size of the file and the amount of data processed in the word cloud generation.

IV. RadCloud Visualization

The RadCloud visualization shows words extracted from different texts or text categories in a merged view. The design of RadCloud is inspired by the RadViz visualization technique [13]: RadViz encodes multi-dimensional points in a 2D projection as a point plot; the dimensions are mapped to equidistant attractors on a circle and points are moved inside the circle towards the attractors according to their values on the dimensions. In RadCloud, the text categories form the dimensions of the data set and we add words instead of points to the circle.

However, words consume considerable space and quickly become unreadable if they overlap. We therefore relax the strict positioning of the original RadViz approach: starting with the most relevant words, our layout algorithm tries to place the words as close as possible to the intended positions, but in an overlap-free layout.

A. Design Decisions

As mentioned above, one possible solution for comparing word clouds would be small multiples: several independent word clouds placed side by side (Figure 1, top). While this solution might provide some insight, it is not very efficient with respect to comparison tasks:

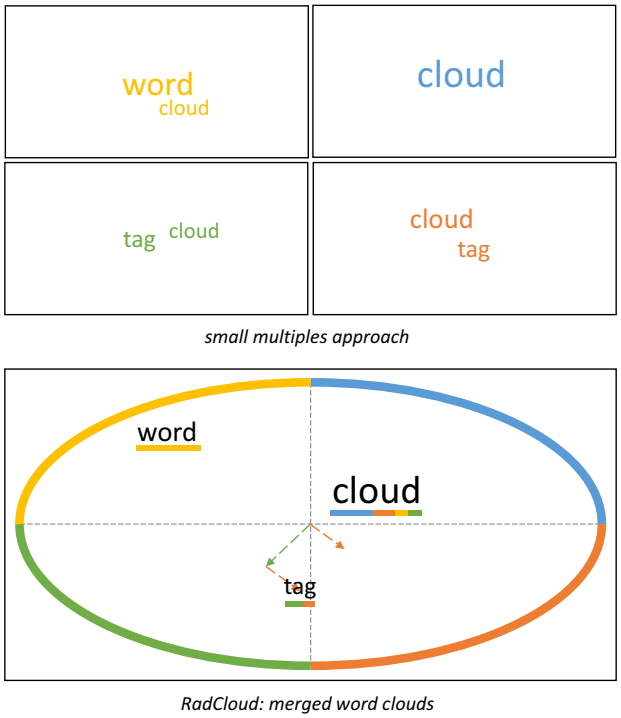

Figure 1. Contrasting a small multiples approach for word cloud comparison (top) to the merged word cloud approach of RadCloud (bottom) based on an illustrating example with three words and four color-coded text categories.

- **Set comparison:** Quick comparisons are problematic in a small multiples view because words are represented several times and need to be matched manually. Consequently, we only allow one representing word in our merged word cloud instead of multiple ones. The font size of each word encodes the maximum relevance value for the word in all categories, whereas its color is derived from the colors of the text categories.
- **Relevance in categories:** Different relevance values for the words in each of the categories lead to different font sizes and layouts in the small multiples. A comparison of word sizes is difficult between multiple representations of the word. To solve this problem in the merged word cloud, we add stacked bar charts to each word in order to indicate how relevant it is in each category.

By joining all representatives of the same word and indicating the relevance with respect to different categories by color-coded bar charts (Figure 1, bottom), we hence already ease the comparison tasks. As an additional step described in the following, we further use the layout of the word cloud to indicate which words are assigned to which categories. This complements the approach: the layout gives a first impression of the category assignment while the bar charts indicate the specific category relevance values.

B. Intended Word Positions

We developed a new spring-based layout algorithm (similar to the work of Olsen et al. [18]) to meet the design goals of the approach. We determine each word position inside the circle by first calculating a normalized relative weight w'^c_i in each category c: Let $C := \{c_1, \ldots, c_n\}$ be a set of categories and W_c the set of words that belong to category c. The set of all words is defined as

$$W := \bigcup_{c \in C} W_c.$$

Then, we obtain weights

$$w^c_i = \begin{cases} 0 & if \quad w_i \notin W_c \\ weight(word_i, category_c) \in \mathbb{R} & if \quad w_i \in W_c \end{cases}$$

For the normalized weight in a category c, we finally get

$$w'^c_i := \frac{w^c_i}{\sum_{k=1}^{|W|} w^c_k}.$$

The circle in which the words are placed is centered around the origin of the coordinate system. The categories are equally distributed along this circle. The individual positions are then obtained by rotating the point around the center starting at the top of the circle. The vectors V_c from the origin to the category points are created as $V_c := \overrightarrow{position(category_c)}$. The category positions and spring vectors are cached and reused as long as the rendering area, the category order, or the visibility do not change. The intended placement for each word is then obtained using

$$w''^c_i := \frac{w'^c_i}{\sum_{k=1}^{|C|} w'^k_i}.$$

For the position of a word, we obtain

$$position(word_i) := \sum_{c=1}^{|C|} (V_c \times w''^c_i).$$

Figure 1 (bottom) illustrates the composition of the intended position by vectors.

C. Corrected Word Positions

RadViz was designed for the representation of small data points. As already mentioned, words require a comparatively large area to be displayed, leading to much overlap, which impairs the readability of a merged word cloud. For this reason, we propose a strategy that places words close to their intended position, while still maintaining a readable representation without overlaps.

Springs are used to model distances between the bounding boxes of the words. When two words overlap, they are moved away from each other by a minimal distance. Yet, the distance is not the only criterion by which a new location for each word is computed: Since the position of a word relative to the circle center is semantically relevant, it should not change too much. Therefore, a value is calculated that represents the entropy of the new position and depends on both the distance and angle in which the word is moved. When a movement is applied, it does not dislocate a single word by the complete distance but distributes it to both involved words.

V. Example

The combined visualization of different documents and document collections allows new application and analysis scenarios that cannot be addressed with traditional word clouds. To show how RadCloud can be used in practice, we applied the approach to a sample data set. In particular, we studied different characteristics of groups of metals. Six categories of metals are discerned according to their chemical characteristics: *alkali metals*, *alkaline earth metals*, *transition metals*, *actinides*, *lanthanides*, and *poor metals*. Wikipedia articles of all 92 metals belonging to the six categories were taken as a text base. Figure 2 shows the RadCloud visualization generated from the texts using 100 words per category.

The overall word cloud is shifted towards the upper left, as transition metals (pink) form the largest group of metals (38 metals) and hence have the largest text base. Not surprisingly, general words from the domain of metals, such as *element*, *isotopes*, or *chemical* as well as *metal*, can be found in the center region of the visualization: the stacked bars show that they are shared by nearly all text categories, while the large font size further indicates their relevance.

On the contrary, some words are placed in close proximity to the attraction points of the six categories—these words appear exclusively in one category. Among them are names of specific metals, such as *sodium*, *lithium*, and *rubidium* (alkali metals) or *calcium*, *berylium*, or *magnesium* (alkaline earth metals).

In addition to general terms and concrete metals, there are also some words that point to specific characteristics of groups of metals. For instance, words exclusively assigned to a single category describe those characteristics:

- The word *oxygen* attached to alkali metals points to their high reactivity with oxygen.
- The word *bone* is related to alkaline earth metals because calcium salts are important for the structure of a bone.
- The word *berkeley* is central for actinides as research facilities in Berkeley, CA, USA played an important role in their discovery and investigation (one is named *berkelium*).

Furthermore, one can observe that many words in the RadCloud are related to radioactivity: *half-life*, *radioactive*, *radiation*, *alpha*, *beta*. Some of these individual words are not related to all categories, but taken together, they cover all categories of metals, as radioactivity is not limited to a certain category of metals.

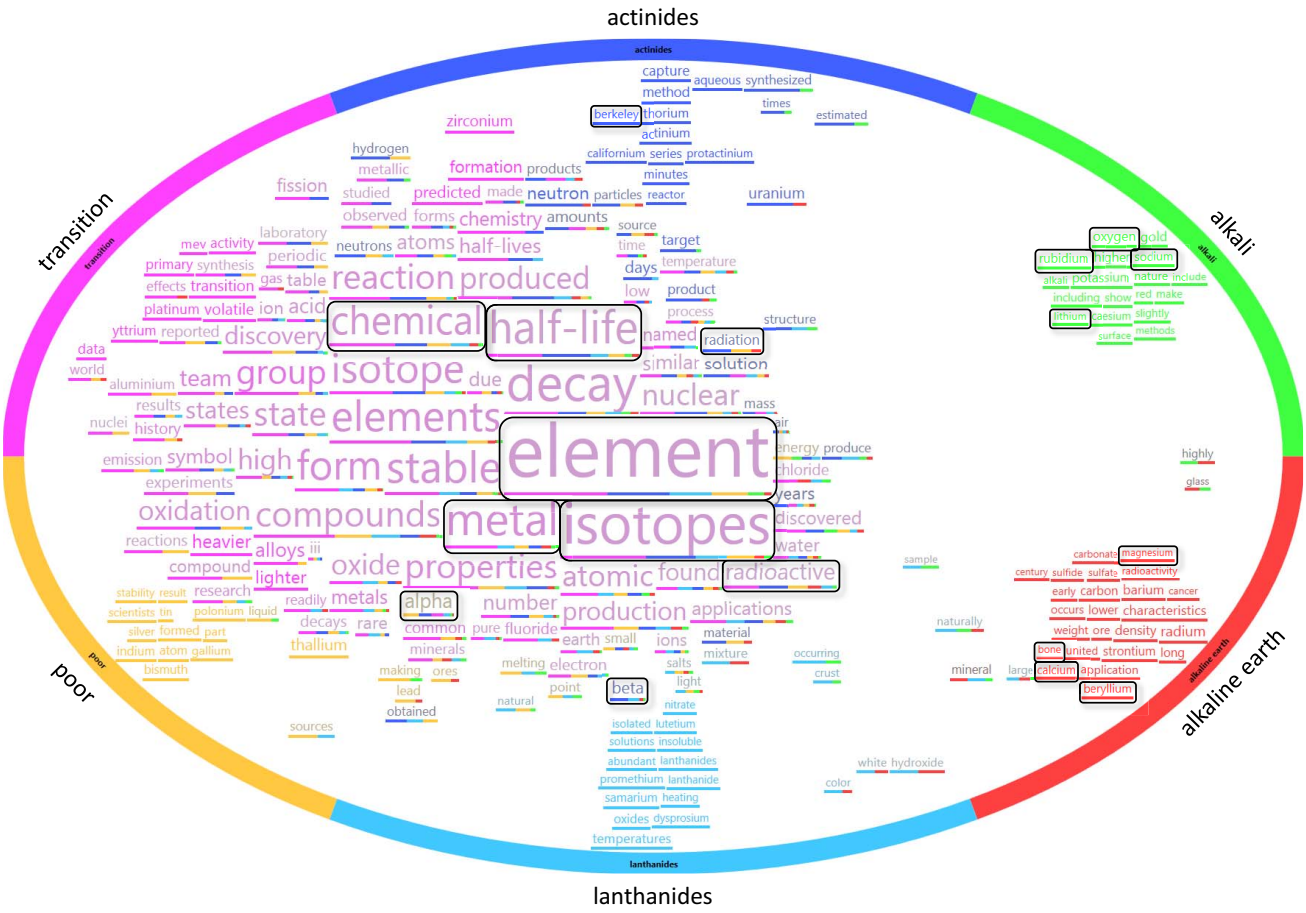

Figure 2. Annotated RadCloud generated from 100 words per category for six different groups of metals derived from Wikipedia articles; black frames indicate the words discussed in the paper.

VI. Conclusion

We have proposed RadCloud, an approach to visually merge multiple word clouds for different documents and document collections. In contrast to separate juxtaposed word clouds, each word is represented only once, i.e. users do not need to search across different word clouds in order to compare the relevance of a word.

Inspired by the RadViz approach, the position of a word is chosen according to its relevance in the different text categories employing a spring-based model in a radial layout. To avoid overlapping words, we correct the positions but move the words as little as possible.

While the position is a first indicator for the relevance of a word with regard to a specific text category, the relevance vector is additionally visualized as a stacked bar below the word. Unique colors are used to identify the text categories. The RadCloud visualization has been connected to an elaborate text processing approach and implemented in an interactive tool. The application of the approach is illustrated by an example.

References

[1] IR Multilingual Resources at UniNE. http://members.unine.ch/jacques.savoy/clef/, 2014.

[2] M. Burch, S. Lohmann, D. Pompe, and D. Weiskopf. Prefix tag clouds. In *Proceedings of the 17th International Conference on Information Visualisation*, IV '13, pages 45–50. IEEE, 2013.

[3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

[4] Y.-X. Chen, R. Santamaría, A. Butz, and R. Therón. Tagclusters: Semantic aggregation of collaborative tags beyond tagclouds. In *Proceedings of the 10th International Symposium on Smart Graphics*, SG '09, pages 56–67. Springer, 2009.

[5] C. Collins, F. B. Viégas, and M. Wattenberg. Parallel Tag Clouds to explore and analyze faceted text corpora. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, VAST '09, pages 91–98. IEEE, 2009.

[6] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu. Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, 30(6):42–53, 2010.

[7] J. Feinberg. Wordle. In J. Steele and N. Iliinsky, editors, *Beautiful Visualization*, pages 37–58. O'Reilly, 2010.

[8] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACL, 2005.

[9] K. Fujimura, S. Fujimura, T. Matsubayashi, T. Yamada, and H. Okuda. Topigraphy: visualization for large-scale tag clouds. In *Proceedings of International Conference on World Wide Web*, WWW '08, pages 1087–1088, 2008.

[10] P. Gambette and J. Véronis. Visualising a text with a Tree Cloud. In *Proceedings of IFCS Biennial Conference and 33rd Annual Conference of the Gesellschaft für Klassifikation e.V.*, pages 561–569. Springer, 2010.

[11] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies*, pages 25–28, 2006.

[12] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl. Word cloud explorer: Text analytics based on word clouds. In *Proceedings of the 2014 47th Hawaii International Conference on System Sciences*, HICSS '14, pages 1833–1842. IEEE, 2014.

[13] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *Proceedings of the 8th Conference on Visualization '97*, VIS '97, pages 437–441. IEEE, 1997.

[14] O. Kaser and D. Lemire. Tag-Cloud Drawing: Algorithms for cloud visualization. In *WWW '07 Workshop on Tagging and Metadata for Social Information Organization*, 2007.

[15] K. Koh, B. Lee, B. Kim, and J. Seo. ManiWordle: Providing flexible control over Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1190–1197, 2010.

[16] B. Lee, N. H. Riche, A. K. Karlson, and S. Carpendale. SparkClouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, 2010.

[17] S. Lohmann, M. Burch, H. Schmauder, and D. Weiskopf. Visual analysis of microblog content using time-varying co-occurrence highlighting in tag clouds. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 753–756. ACM, 2012.

[18] K. A. Olsen, R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: The VIBE system. *Information Processing and Management*, 29(1):69–81, 1993.

[19] F. V. Paulovich, F. M. B. Toledo, G. P. Telles, R. Minghim, and L. G. Nonato. Semantic wordification of document collections. *Computer Graphics Forum*, 31(3):1145–1153, 2012.

[20] J. Schrammel, M. Leitner, and M. Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2037–2040, 2009.

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

[22] M. Stefaner. Visual tools for the socio-semantic web. Master thesis, University of Applied Sciences Potsdam, 2007.

[23] H. Strobelt, M. Spicker, A. Stoffel, D. Keim, and O. Deussen. Rolled-out Wordles: A heuristic method for overlap removal of 2D data representatives. *Computer Graphics Forum*, 31(3):1135–1144, 2012.

[24] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. ManyEyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[25] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar. What's being said near "Martha"? Exploring name entities in literary text collections. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, VAST '09, pages 107–114. IEEE, 2009.

[26] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma. Semantic-preserving word clouds by seam carving. *Computer Graphics Forum*, 30(3):741–750, 2011.

[27] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: Interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization on Computer Graphics*, 16(6):1109–1118, 2010.