The ICE-Map Visualization

Kai Eckert
kai@informatik.uni-mannheim.de

Technical Report TR-2011-003
Department of Computer Science
University of Mannheim, Germany

December 7, 2011

Abstract

In this paper, we describe in detail the Information Content Evaluation Map (ICE-Map Visualization, formerly referred to as IC Difference Analysis). The ICE-Map Visualization is a visual data mining approach for all kinds of concept hierarchies that uses statistics about the concept usage to help a user in the evaluation and maintenance of the hierarchy. It consists of a statistical framework that employs the the notion of information content from information theory, as well as a visualization of the hierarchy and the result of the statistical analysis by means of a treemap.

1 Introduction

In 1854, there was a severe Cholera outbreak in London in the Soho district. At this time, people generally believed that Cholera was caused by polluted air (miasma theory). John Snow, a physicist, questioned this theory and tried to find evidence for another source for the Cholera, particularly the drinking water. So he investigated the Cholera cases carefully and gathered a lot of data about them. He drew a map of the affected area and marked every fatal case with black bars (Figure 1).

On this map, it can be seen that the cases are scattered around the Broad Street and based on the distribution, John Snow had the suspicion that the water pump in the Broad Street could be the source. He convinced the district council to disable the pump and subsequently, the Cholera cases decreased.

This is the very condensed version of the story that is often called in various slightly modified versions as the "invention" of visual data mining, i.e. the

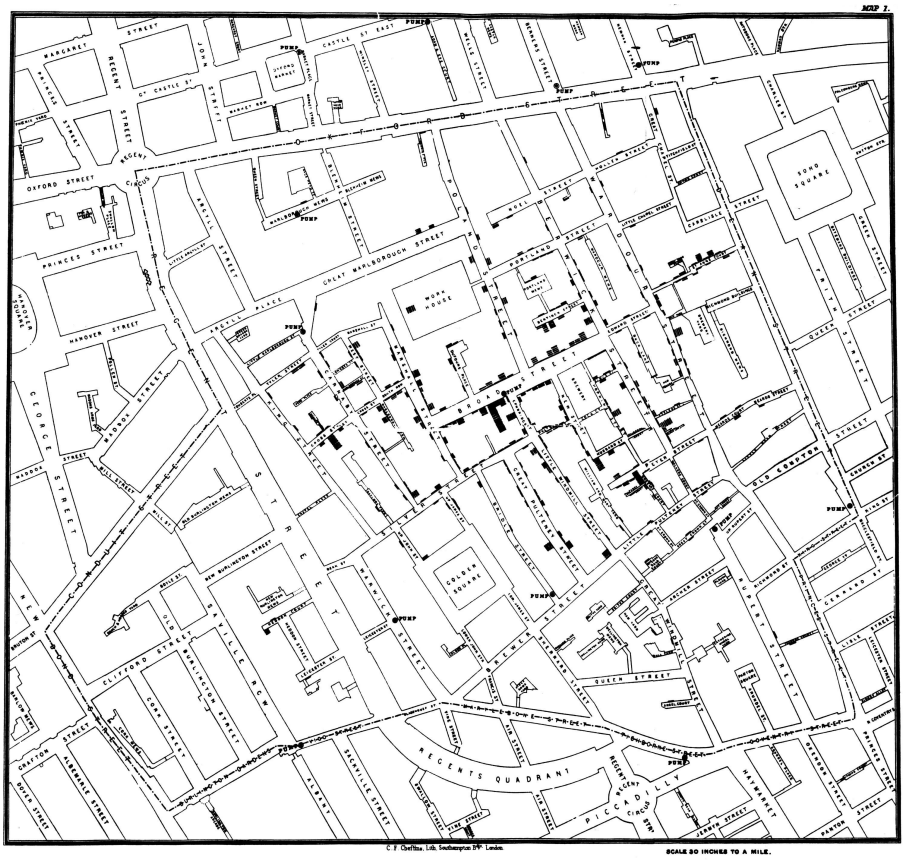

Figure 1: Original map made by John Snow in 1854. Cholera cases are highlighted in black.

analysis of data and the identification of correlations by means of a proper visualization. In this sense, it became a myth[1] (McLeod, 2000).

Visual data mining (VDM) is not just the visualization of data, it usually refers to a whole process that involves the user to interpret the visualization of the data and adapt it interactively to discover interesting correlations or facts that are otherwise not visible and hard to recognize. Applications of visual data mining range from the search for specific information – also in retrieval settings to fulfill a specific need – to the browsing of huge amounts of data to find interesting aspects. The interaction with the data and the visualization plays an important role; the visualization system has to help the user to navigate through the data. This combination of visualization and interaction is solely possible by means of modern computers with high resolution graphical displays.

VDM approaches can be found in various systems and for various purposes. For instance, Fluit, Sabou, and Harmelen (2005) present three different applications that use the Cluster Map technology, an interactive visualization for overlapping clusters. DaCosta and Venturini (2006) use the concept of points of interest for the purpose of VDM on numeric or symbolic data. G. Smith et al. (2006) introduce FacetMap, an interactive visualization, primarily as a means to organize and retrieve data from various heterogeneous sources.

Shneiderman (1996, p. 337) introduces a mantra for this kind of VDM: "Overview first, zoom and filter, then details-on-demand."

Overall, he identified seven high-level tasks for VDM:

- Overview: Gain an overview of the entire collection.
- Zoom : Zoom in on items of interest.
- Filter: filter out uninteresting items.
- Details-on-demand: Select an item or group and get details when needed.
- Relate: View relationships among items.
- History: Keep a history of actions to support undo, replay, and progressive refinement.
- Extract: Allow extraction of sub-collections and of the query parameters.

[1] In fact, John Snow was not the first one to use maps for data visualization and the drawing of the map was actually only one means for his investigations (Koch, 2004) – see also the original report (Snow, 1855). Nevertheless, the "Ghost Map" is a very nice example for the power of a proper visualization. The story about John Snow and the Cholera outbreak is the central theme of a novel called "The Ghost Map" (Johnson, 2006).

2 Foundations

The ICE-Map Visualization is a VDM approach that follows the mantra of Shneiderman and is specifically designed for the purpose of maintenance and use of concept hierarchies in various settings. In this paper, we use the following definitions:

Definition 2.1 (Concept) *A concept represents a real world object and/or abstract entity like a knowledge concept. As a representation, it usually consists of a description that defines the scope of the concept and one or more labels to name (and also define) the concept.*

Definition 2.2 (Knowledge Organization System) *A Knowledge Organization System (KOS) is a structured model of concepts that is used to represent and organize knowledge.*

Definition 2.3 (Concept Hierarchy) *A concept hierarchy is a knowledge organization system that provides at least one relationship between the concepts that leads to a hierarchical structure. A common example for such a relationship is "broader than", respectively "narrower than", but there are also others, like "is a"/"has subclass" or "part of"/"has part".*

Essential for building a hierarchy is the ***transitivity*** *of the relationship: if A is broader than B and B is broader than C, A has also to be broader than C or the organization in a hierarchy would be counter-intuitive.*

During the creation, maintenance, and the actual use of concept hierarchies, there are several tasks to be done and several questions to be answered that require a sound knowledge of the concept hierarchy. Therefore, in a dynamic setting, where new concept hierarchies have to be chosen and deployed, as well as during the maintenance of existing concept hierarchies, a proper tool support is required.

As an motivating example, consider the use of an automatic indexing system. Naturally, one is interested in the evaluation of the indexing results, but the evaluation is not trivial for thesauri with thousands of concepts and thousands of documents to be indexed. The ICE-Map Visualization is developed for exactly this scenario: The concept use is visualized to allow for a proper and intuitive evaluation of the indexing result.

Literature on thesaurus creation and maintenance mentions a number of tasks that might be necessary including the following taken from (Kuhlen, Seeger, & Strauch, 2004, pp. 151-153):

1. Adaptation of the concept hierarchy to changes in the vocabulary of the domain of interest by means of adding of new terms or concepts,

2. deletion and/or merging of rarely used concepts,
3. splitting, extension or restriction of extensively used concepts,
4. review of the hierarchical structure to avoid extensive subclassing and

Depending on the intended use of the KOS, there might be more tasks. For example, if the KOS should be used for automatic indexing, the following task has to be added:

- Identification of problematic concepts for the indexing software, i.e. concepts that are erroneous assigned or missing.

The first task is a different issue, that we addressed for instance in (Meusel, Niepert, Eckert, & Stuckenschmidt, 2010). The remaining four tasks can be supported by means of the ICE-Map Visualization, which makes it a very universal and powerful tool.

In order to enable a domain expert to carry out these actions, we analyze the thesaurus and detect unbalanced hierarchy structures as well as terms that are more often or less often used in indexing than we would expect. We support this step using a statistical measure that comes along with a proper visualization that makes it easy for the user to spot potential problems.

The ICE-Map Visualization is designed to visualize the usage of concepts in a hierarchy, for example, how often they are used for indexing. Therefore, we further define as follows:

Definition 2.4 (Document) *A document is a single information item or resource, that usually contains textual information and is made available to the IR system with one or more of the following attributes:*

1. *title,*
2. *abstract,*
3. *fulltext (structured or plain text),*
4. *other bibliographic information (creator(s), publisher, year, identifiers, ...),*
5. *links to textual representations and/or descriptions of the content.*

Definition 2.5 (Annotation) *An annotation is the assignment of a concept to a document (Definitions 2.4 and 2.1) for information retrieval purposes, i.e. the result of an indexing process.*

3 Statistical Framework

Table 1 introduces the mathematical notation based on the definitions so far that is used in the remainder of this section.

Symbol	Explanation
c	A concept according to Definition 2.1.
$\text{Children}(c)$	The direct child concepts (narrower concepts) of c.
$\text{Children}^+(c)$	All recursive child concepts (narrower concepts) of c.
$\text{Parents}(c)$	The direct parent concepts (broader concepts) of c. That can be more than one in the case of a polyhierarchy.
$\text{Siblings}(c)$	The sibling concepts of c. In case of multiple parents, the corresponding parent has to be denoted, but we skip this here for simplicity.
$\text{Aset}(c)$	The set of annotations (Definition 2.5) related to concept c.
H	A concept hierarchy according to Definition 2.3. H is a partially ordered set of concepts based on the broader/narrower relationship and forms a (polyhierarchic) tree.
$\text{root}(H)$	The root concept of H, i.e. the only concept c in H for which holds that $\text{Parents}(c) = \emptyset$. Note that we require H to have a single root concept. Otherwise, we introduce an artificial single root concept that becomes the parent of all former root concepts.
$\text{root}(c)$	The root concept of the concept hierarchy H where c belongs to.
Lower case denotes single elements, while upper case denotes sets. Accordingly, functions returning single elements are written lower case, functions returning sets are written upper case.	

Table 1: Symbols and definitions

The usage of a concept c is determined by a weight function $w(c) \in \mathbb{R}_0^+$ that assigns a non-negative, real weight to it. Based on this weight function, we further define:

$$w^+(c) = w(c) + \sum_{c' \in \text{Children}(c)} w^+(c') \tag{1}$$

$w^+(c)$ is a monotonic function on the partial order of the concept hierarchy H, i.e. the value never increases while walking down the hierarchy. This gives the value of the root node a special role as the maximum value of w^+, which we denote as $\hat{w}^+$:

$$\hat{w}^+(c) = w^+(\text{root}(c)) = \max_H w^+(c) \tag{2}$$

Now, we come back to our motivating example. If we use the number of annotations made for a given concept as the weight function $w(c)$, we can calculate the likelihood that a concept is assigned to a random document as follows:

$$L(c) = \frac{w^+(c) + 1}{\hat{w}^+(c) + 1} \qquad L(c) \in (0, 1] \tag{3}$$

The addition of 1 is necessary to allow a value of 0 for $w(c)$. Otherwise, the logarithm of $L(c)$ (cf. Equation 4) would not be defined for $w(c) = 0$.

In information theory, the Information Content or Self-information of an event x is defined as $-\log L(x)$, i.e., the information content of an event is the higher, the more unlikely the event is. Together with a normalizing factor, we get the following definition for the Information Content $IC(c) \in [0, 1]$ of a concept c:

$$IC(c) = \frac{-\log L(c)}{\log(\hat{w}^+(c) + 1)} \qquad \hat{w}^+(c) \neq 0 \tag{4}$$

This is again a monotonic function on the partial order of H and assigns 0 to the root concept and 1 to concepts with $w(c) = 0$.

The ICE-Map Visualization always compares two data sets based on the difference of the information content. Therefore we originally referred to it as IC Difference Analysis, but that way the underlying statistics could be confused with the overall VDM approach. Nevertheless, the basis of the ICE-Map Visualization is the difference of two information content calculations $IC_A(c)$ and $IC_B(c)$ by means of two different weight functions or a weight function applied to two different data sets, e.g. two sets of annotations from two different indexing processes. Accordingly, we define the IC Difference $D_B^A(c) \in [-1, 1]$:

$$D_B^A(c) = IC_A(c) - IC_B(c) \tag{5}$$

The ICE-Map Visualization is usually used to evaluate a data set against some reference, thus we refer to A as "analysis" and B as "base".

4 Weight Functions

The power of the ICE-Map Visualization lies in the possibility to choose arbitrary weight functions for analysis A and base B. In this section, we introduce two simple weight functions that can be used:

$$w_{IC}(c) = |\mathrm{Aset}(c)| \tag{6}$$

If we use the number of annotations for a given concept as $w(c)$, Equation 4 corresponds[2] to the information content of a concept, as introduced by Resnik (1995). When Equation 6 is used, it has to be denoted somehow, which annotation set is actually used, especially, when two different sets are to be compared.

Sometimes, one might want to use the ICE-Map Visualization to evaluate data sets without the availability of a reference set. Especially during the maintenance of a concept hierarchy this will happen quite often, as usually no two sets of annotations are available for a given document base. For this reason, we introduce the second weight function that can be used as a heuristic to determine the expected information content of a concept:

$$w_{IIC}(c) = |\text{Children}(c)| \tag{7}$$

Equation 7 is based on the assumption that more common concepts should have a lesser information content than more specific concepts. The use of Equation 7 leads to the so called Intrinsic Information Content (IIC), i.e. an information content that is determined only by means of the thesaurus structure itself, as introduced by Seco, Veale, and Hayes (2004).

Depending on the applied weight functions, we can create various configurations for the ICE-Map Visualization, leading to different applications:

$D_{ICb}^{ICa}(c)$ This way, the a set of annotations can directly be compared to a reference set of – usually intellectually created – annotations. This measure shows for example deviations between manually and automatically assigned concepts and therefore directly points to potential problems in the automatic indexing process.

$D_{IIC}^{ICa}(c)$ With this configuration, we can monitor an indexing system without the need of a reference set.

$D_{IIC}^{ICb}(c)$ This configuration is very useful to gain an overview on the focus of a document base, based on intellectually assigned concepts, as well as to gain an understanding of the characteristics of the reference set or the underlying KOS.

$D_{ICa}^{ICa}(c)$ With two different indexed document sets, we can compare both sets, for example to compare the foci of two different libraries.

5 Visualization

The statistical framework is only one half of the ICE-Map Visualization. While it can be used independently of the visualization to calculate the IC Difference

[2]Beside the normalization and the addition of 1 to deal with zero values.

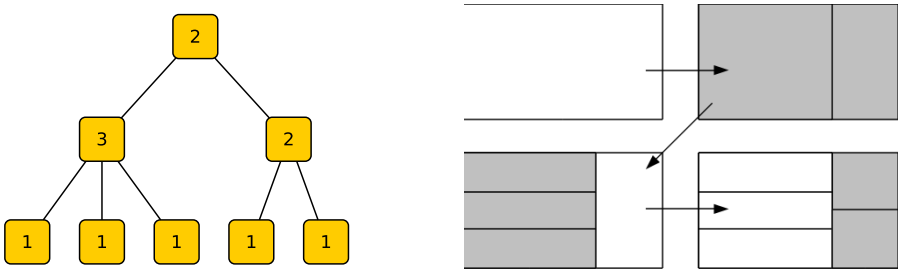

Figure 2: Original "slice-and-dice" layout

for one concept, the main purpose is to provide the user with the big picture of a full analysis of a concept hierarchy.

A major challenge in supporting KOS maintenance is to provide adequate tool support that guides the user to potential problems in a KOS based on the measures described above. In particular we have to find a way to provide the user with a view on the concept hierarchy that encodes the overall structure of the thesaurus or selected parts of it and the evaluation results for the different concepts in the thesaurus.

The ICE-Map Visualization uses a treemap to visualize the concept hierarchy together with the results of the analysis. The treemap visualization was developed by Shneiderman (1992) in the early 1990s, originally with the purpose to get an overview of disc usage of a particular hard drive. Shneiderman needed a compact representation of its directory structure, showing additional information like file size and file type in one view.

According to Shneiderman, treemaps are a representation designed for human visualization of complex traditional tree structures: arbitrary trees are shown with a 2-d space-filling representation. Consider a tree with weight or size information attached to each node and a 2-d space with corners (x_1, y_1) and (x_2, y_2). For each child c_i of the root node r, a partition of the space along the x-axis is calculated. For the first partition, this reads as

$$x_3 = x_1 + \left(\frac{|c_1|}{|r|} \right) (x_2 - x_1) \tag{8}$$

with $|c_1|$ as the size of child node 1 and $|r|$ as the size of the root node. For the next level, the corresponding partition is partitioned again along the y-axis, then again on the x-axis and so on (Figure 2). Shneiderman called this approach the "slice-and-dice" algorithm. Since then, a lot of different implementations and optimizations have been presented, e.g. by Shneiderman and Wattenberg (2001) or Bederson, Shneiderman, and Wattenberg (2002).

6 Squarified Layout

In our implementation, we use the squarified layout, as presented by Bruls, Huizing, and Wijk (2000). In the description of the algorithm below, we use the following conventions:

We want to layout the children $c \in C$ of a given parent concept p. Therefore, we want to determine the dimensions (width, height) and the position (x,y) of the rectangle that is occupied by each concept. We denote them with c_w, c_h, c_x, and c_y, respectively. Note that we regard c and p as compound objects, containing the dimension and position information denoted by the subscript.

We know the dimensions of the area that can be used to layout the child concepts, p_w and p_h. For each concept c, we can calculate its area c_a based on a weight function $w(c) \neq 0$ as a fraction of the area of the parent concept p:

$$c_a = p_w \cdot p_h \cdot \frac{w(c)}{w(c) + \sum_{i \in \text{Siblings}(c)} w(i)} \tag{9}$$

The general idea of the squarified layout is to split the children into several rows that are laid out one after the other. Each row is placed in the lower, left corner of the remaining area and the rectangles of the concepts are assembled horizontally, if the remaining area is higher than wide, and vertically, if the remaining area is wider than high. In the first case, a row uses the full width of the remaining area, in the latter the full height.

We can calculate the width and height of every concept c in a row R, as well as its position, i.e. the coordinates of its lower left corner, based on the rectangle s of the remaining free area. First, we introduce the calculation of c_w and c_h under the assumption that the row is laid out horizontally with a given *width* and with relative positioning, i.e. the lower left corner of the row is $(0, 0)$.

CALCULATE-ROW(R, *width*)

```
1  area = Σ_{c∈R} c_a                    // cf. Equation 9.
2  height = area/width
3  x = 0
4  for c ∈ R
5      c_w = width · (c_a/area)
6      c_h = height
7      c_x = x
8      x = x + c_w
9      c_y = 0
10 return height
```

Note that CALCULATE-ROW returns the *height* of the calculated row. This is used in the following procedure, where a row R is actually placed within a free area s. PLACE-ROW adheres to the above mentioned strategy and returns the remaining free area after the placement of the new row.

PLACE-ROW(R, s)

```
1  width = min(s_w, s_h)
2  height = CALCULATE-ROW(R, width)
3  if s_w > s_h                          // Distinction between horizontal and
4       ROTATE-ROW(R)                    // vertical layout, see text above.
5       s'_w = s_w − height
6       s'_h = width
7       s'_x = s_x + height
8       s'_y = s_y
9  else s'_w = s_w
10      s'_h = s_h − height
11      s'_x = s_x
12      s'_y = s_y + height
13 SHIFT-ROW(R, s)
14 return s'
```

The rotation[3] – if the row has to be layed out vertically – and shift of the row are implemented as follows, using simple vector arithmetic:

ROTATE-ROW(R)

```
1  for c ∈ R
2       SWAP(c_w, c_h)
3       SWAP(c_x, c_y)
4  return
```

SHIFT-ROW(R, s)

```
1  for c ∈ R
2       c_x = c_x + s_x
3       c_y = c_y + s_y
4  return
```

The remaining question is: How should the children be distributed to the single rows? The heuristic used in this case is as follows: Sort the children by their size in descending order and then start adding them to a row R. Then calculate the "badness" of the row based on the worst aspect ratio of the concepts in the row:

$$\text{badness}(R) = \begin{cases} \max_{c \in R} |c_w / c_h - 1| & R \neq \emptyset \\ \infty & R = \emptyset \end{cases} \tag{10}$$

If the addition of a concept would increase the badness, do not add it and instead start a new row. This leads to the following procedure for a parent concept p and its children C:

[3]The rotation is that simple because the position of the concept is only relative at the time of the invocation, i.e. it is a rotation around $(0, 0)$.

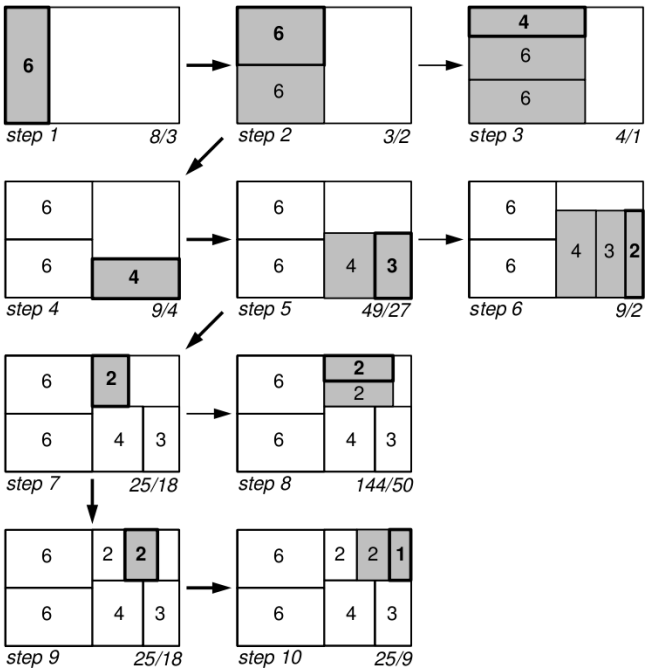

Figure 3: Squarified Layout (Source: Bruls, Huizing, and Wijk, 2000)

LAYOUT-CHILDREN(C, p)

```
1  Sort C by c_a decreasing                    // cf. Equation 9.
2  R = T = ∅                                   // Initialize rows, T means temporary.
3  s = COPY(p)                                 // Start with the area of p.
4  for c ∈ C
5      ADD(T, c)
6      CALCULATE-ROW(T, min(s_w, s_h))
7      if badness(T) > badness(R)              // Check, if badness is increased.
8          s = PLACE-ROW(R, s)                 // Place row and
9          R = T = ∅                           // start a new one.
10         ADD(T, c)                           // Prepare T for next row.
11     ADD(R, c)                               // Extend the row and continue.
12 if R ≠ ∅
13     PLACE-ROW(R, s)                         // Place remaining concepts, if any.
14 return
```

Figure 3 (Bruls et al., 2000) illustrates the algorithm for one concept with child concepts having the weights $(6, 6, 4, 3, 2, 2, 1)$.

With LAYOUT-CHILDREN, we can now recursively layout the whole treemap. The drawing of the treemap gives us two degrees of freedom that can be used

to visualize information beside the hierarchical structure. One is represented by the size of the concepts, the other by its color.

We experimented with various combinations of metrics to determine the size and color weights of a concept. It turned out that the size should usually not be used to visualize aspects other than the hierarchy, because otherwise we would not get a stable visualization of the hierarchy that does not change its layout if another analysis on the concept usage is performed.

The most convenient weight function for the size is based on the number of children of a concept, either only the direct children (Equation 7) or with all subchildren (Equation 1 with Equation 7 as internal weight function). Usually the latter is to be preferred, as this way the space is evenly distributed between all the concepts of the hierarchy and thus uses the space optimally to view as much concepts as possible. In any case, some positive value has to be added to the weight function to prevent zero values for concepts without children.

The color is determined by the result of the analysis that is performed on the concept hierarchy. The ICE-Map Visualization uses the IC difference with arbitrary weight functions. In the default setup, the weight between -1 and 1 is mapped to a color range from red (-1) over white (0) to blue (1). The lower the information content of a concept is, the higher is the underlying weight function. This way, the treemap can be interpreted as a temperature map, with red areas indicating "hot" areas regarding the usage (or whatever is used as weight function) and blue ones "cold" areas, compared to the chosen reference.

7 Implementation

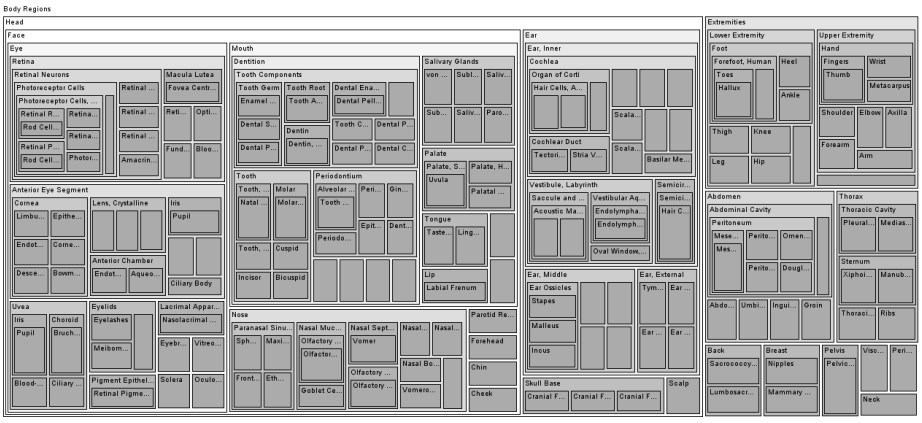

Figure 4: Treemap of the Concept "Body Regions" from MeSH.

Figure 4 shows the treemap of a part of the MeSH thesaurus (Body Regions), where each area represents a concept in the thesaurus. As Bruls et al. (2000)

point out, a drawback of the treemap visualization in general and especially the squarified layout is that it is not easy to recognize the underlying hierarchy. They propose the use of a profiled border, in combination with a cushion visualization (Wijk & Wetering, 1999). Our reference implementation of the ICE-Map Visualization uses nested areas with line borders and a written title on top of each concept – provided there is enough space; otherwise, the title is omitted. In our experiments, we found this very convenient and usually there is no problem to see and understand the nested structure of the underlying hierarchy.

Nevertheless, the treemap visualization requires some time for the user to get familiar with. Thus, the reference implementation introduces further means to improve the usability, following the above mentioned mantra: "Overview first, zoom and filter, then details-on-demand."

First of all, the treemap visualization itself is highly interactive. By double-clicking on a concept in the treemap the user can zoom into the hierarchy. A double-click on the top concept zooms out again. A major drawback of treemaps is the possibility for the user to lose the orientation in the hierarchy as the visualization can not provide information about the environment of the currently selected top concept, when zooming in.

We deal with this problem in two ways (Figure 5): First, we provide a root-line above the treemap visualization, that shows the path from the top of the hierarchy to the currently shown concept. The concepts in the root-line are colored accordingly. A click on a concept in the root-line directly zooms out to the concept.

Second, the treemap is combined with a hierarchical common treeview. This allows interactive navigation through the hierarchy without losing the orientation. The selection of a concept in the treeview leads to a selection of the concept in the treemap and vice versa. For a selected concept, additional information about the concept is provided in a pop-up box.

The colors of the visualization can be adjusted by the slider below the treemap. This way, the contrast can be improved by narrowing the color range to smaller values of the analysis result. Additionally, the balance between red and blue can be adjusted. The latter can be used to set the color of the top concept to white and thus visualize the subconcepts as if the current top concept would be the root of the hierarchy.

8 Related Work

To the best of our knowledge, no one ever used such a combination of statistical analysis and the treemap visualization to perform visual datamining on concept hierarchies. However, there are several aspects of our work where related

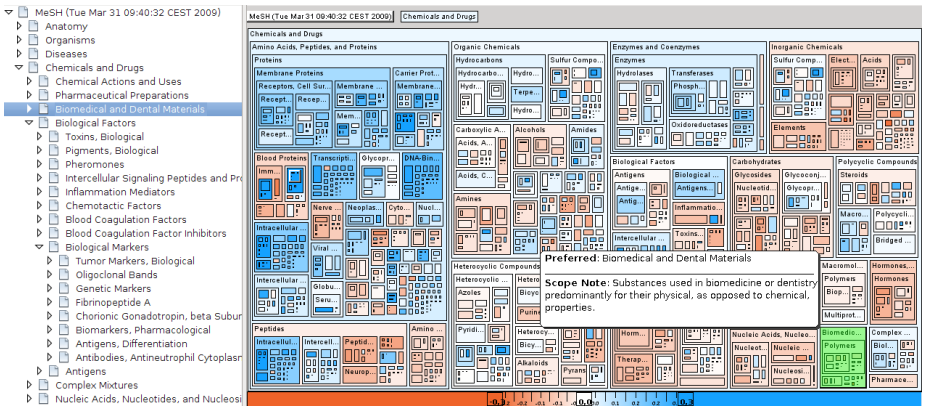

Figure 5: Reference implementation of the ICE-Map Visualization

approaches exist. The treemap visualization itself is widely used, especially to visualize large hierarchical datasets. For example, M. Smith and Fiore (2001) employed it to visualize Usenet newsgroups. Calmet and Daemi (2004a, 2004b) evaluate ontologies using the Kullback-Leibler Divergence, which is widely used in information theory and defined as follows:

$$D_{KL}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)} \tag{11}$$

This is a measure of the differences between two probability distributions p and q and as such related to Equation 5. The authors use the Kullback-Leibler Divergence to get an overall measure of the thesaurus suitability, instead of evaluating a single concept. Rayson and Garside (2000) use a log-likelihood approach to compare text copora and show that it can be used to determine key terms in a corpus which distinguishes it from the reference corpus.

9 Conclusion

In this paper, we described the technical background of the ICE-Map Visualization. Based on this description, everyone should be able to implement it. We do not present any evaluation or usage examples here, you can find them in the papers where we employed the ICE-Map Visualization (cf. Section 10). Our reference implementation is part of SEMTINEL, an analysis and visualization framework for concept hierarchies that is available free and open source[4].

[4] http://www.semtinel.org

10 Acknowledgements

The ICE-Map Visualization was developed as part of my dissertation. While I describe it in this paper for the first time with the statistical framework in a generalized form, the visualization itself exists now for over four years. During the development and especially regarding the possible usage scenarios I was supported by my colleagues Magnus Pfeffer and Heiner Stuckenschmidt. Together, we published several papers: The ICE-Map Visualization was first presented at the Fourth International Conference on Knowledge Capture (K-CAP 2007) as IC Difference Analysis where it was granted the best paper award (Eckert, Stuckenschmidt, & Pfeffer, 2007). In (Eckert, Stuckenschmidt, & Pfeffer, 2008), we published an extended description of the methodology on the use case of the evaluation of automatic indexing results. In (Pfeffer, Eckert, & Stuckenschmidt, 2008), the methodology was adapted to classification systems and automatic classification. In (Eckert, Hänger, & Niemann, 2009), we evaluated tagging results and compared them to intellectual indexing and automatic indexing using the ICE-Map Visualization.

References

Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Trans. Graph.*, *21*(4), 833-854. Available from http://hcil.cs.umd.edu/trs/2001-18/2001-18.pdf

Bruls, M., Huizing, K., & Wijk, J. J. van. (2000). Squarified treemaps. In *Joint Eurographics and IEEE TCVG Symposium on Visualization, IEEE Computer Society* (pp. 33–42). Available from http://www.win.tue.nl/~vanwijk/stm.pdf

Calmet, J., & Daemi, A. (2004a). *Assessing Conflicts in Ontologies* (Tech. Rep.). IAKS Calmet, University Karlsruhe (TH), Germany. Available from http://avalon.ira.uka.de/iaks-calmet/papers/WSEAS_2004.pdf

Calmet, J., & Daemi, A. (2004b). *From entropy to ontology* (Tech. Rep.). Institute for Algorithms and Cognitive Systems (IAKS), University of Karlsruhe (TH), Germany. Available from http://iaks-www.ira.uka.de/calmet/papers/AT2AI4.pdf

DaCosta, D., & Venturini, G. (2006). An interactive visualization environment for data exploration using points of interest. In *Advanced Data Mining and Applications, Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006, Proceedings* (p. 416-423).

Eckert, K., Hänger, C., & Niemann, C. (2009). Tagging and Automation - Challenges and Chances for Academic Libraries. *Library Hi Tech*, *27*(4). Available from http://dx.doi.org/10.1108/07378830911007664

Eckert, K., Stuckenschmidt, H., & Pfeffer, M. (2007). Interactive Thesaurus Assessment for Automatic Document Annotation. In *Proceedings of The*

Fourth International Conference on Knowledge Capture (K-CAP 2007), Whistler, Canada.

Eckert, K., Stuckenschmidt, H., & Pfeffer, M. (2008). Semtinel: Interactive Supervision of Automatic Indexing. In *JCDL '08: Proceedings of the 2008 conference on Digital libraries.* Pittsburgh, PA, USA: ACM, New York.

Fluit, C., Sabou, M., & Harmelen, F. van. (2005). Ontology-based Information Visualisation: Towards Semantic Web Applications. In V. Geroimenko (Ed.), *Visualising the Semantic Web (2nd edition).* Springer Verlag.

Johnson, S. (2006). *The Ghost Map - The Story of London's Deadliest Epidemic - and How It Changed the Way We Think about Disease, Cities, Science, and the Modern World.* Riverhead.

Koch, T. (2004). The Map as Intent: Variations on the Theme of John Snow. *Cartographica*, *39*(4), pp. 1-13.

Kuhlen, R., Seeger, T., & Strauch, D. (Eds.). (2004). *Grundlagen der praktischen Dokumentation und Information, Band 1.* Saur.

McLeod, K. S. (2000). Our sense of Snow: the myth of John Snow in medical geography. *Social Science & Medicine*, *50*(7-8), 923 - 935. Available from http://www.sciencedirect.com/science/article/B6VBF-3YDG0NP-3/2/4aa97b73f3392c7b3110f84ba28decb7

Meusel, R., Niepert, M., Eckert, K., & Stuckenschmidt, H. (2010). Thesaurus Extension using Web Search Engines. In *International Conference on Asia-Pacific Digital Libraries (ICADL), Brisbane, Australia.*

Pfeffer, M., Eckert, K., & Stuckenschmidt, H. (2008). Visual Analysis of Classification Systems and Library Collections. In *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries, Aarhus, Denmark* (p. 436-439). Springer, Heidelberg. Available from http://dx.doi.org/10.1007/978-3-540-87599-4_57

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the workshop on comparing corpora - volume 9* (pp. 1–6). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/1117729.1117730

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95).* Available from http://arxiv.org/pdf/cmp-lg/9511007

Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence* (p. 1089-1090). Valencia, Spain. Available from http://eden.dei.uc.pt/~nseco/ecai2004b.pdf

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, *11*(1), 92–99.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)* (pp. 336–343). IEEE Computer Society,

Washington, DC, USA. Available from http://www.cs.uta.fi/~jt68641/infoviz/The_Eyes_Have_It.pdf

Shneiderman, B., & Wattenberg, M. (2001). *Ordered Treemap Layouts.* Online. Available from ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.pdf

Smith, G., Czerwinski, M., Meyers, B., Robbins, D., Robertson, G., & Tan, D. S. (2006). FacetMap: A Scalable Search and Browse Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *12*(5), 797-804.

Smith, M., & Fiore, A. (2001). Visualization components for persistent conversations. In *Proceedings of the SIG-CHI on Human factors in computing systems* (p. 136-143). Available from http://research.microsoft.com/research/coet/Communities/chi2001/paper.pdf

Snow, J. (1855). Report on the Cholera Outbreak in the Parish of St. James, Westminster, during the Autumn of 1854. In The Cholera Inquiry Committee (Ed.), (chap. Dr. Snow's Report). Churchill, London. Available from http://johnsnow.matrix.msu.edu/work.php?id=15-78-55

Wijk, J. J. van, & Wetering, H. van de. (1999). Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'99), San Francisco, October 25-26, 1999.*