Document Cards: A Top Trumps Visualization for Documents

Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel,
Daniel A. Keim, and Oliver Deussen

Fig. 1. Document Cards help to display the important key terms and images of a document in a single compact view.

Abstract—Finding suitable, less space consuming views for a document's main content is crucial to provide convenient access to large document collections on display devices of different size. We present a novel compact visualization which represents the document's key semantic as a mixture of images and important key terms, similar to cards in a top trumps game. The key terms are extracted using an advanced text mining approach based on a fully automatic document structure extraction. The images and their captions are extracted using a graphical heuristic and the captions are used for a semi-semantic image weighting. Furthermore, we use the image color histogram for classification and show at least one representative from each non-empty image class. The approach is demonstrated for the IEEE InfoVis publications of a complete year. The method can easily be applied to other publication collections and sets of documents which contain images.

Index Terms—document visualization, visual summary, content extraction, document collection browsing.

1 Introduction

Nowadays, large document collections, such as research paper corpora and news feeds, grow at high rate. Many of these documents contain text and images for describing facts, methods, or telling stories. It is an exhaustive task for a user to get an overview of of a larger collection. To overcome this problem, it is common to represent a document in a different way. For instance, search engines usually show the title of a document together with a small context of the query terms. With this representation a user has to read only a portion of the text and is focused on the relevant parts of the documents, which efficiently allows him or her to differentiate between relevant and non-relevant documents. While this representation is efficient to browse through search results, it is not capable to give a quick overview of a document or even a whole document collection.

We introduce a novel approach for a compact visual representation of a document, called Document Card (DC) that makes use of important key terms and important images (see Fig. 1). This representation adopts the idea of top trumps game cards, on which expressive pictures and facts provide a combined overview of an object, such as cars. By using terms as well as images, we maintain the informative value of texts and combine it with the descriptive nature of images in one view. Our visualization aims at compact size so it can scale to handle a large number of documents on display devices of different resolutions.

In Section 2 we give an overview of other work and techniques related to our approach. Our approach is introduced in Section 3. Section 4 contains a detailed description of our technique to create Document Cards. In Section 5 we present an application of Document Cards to a corpus of scientific documents. We conclude with future work in Section 6.

- *Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Daniel A. Keim, and Oliver Deussen are with the University of Konstanz, E-mails: {hendrik.strobelt, daniela.oelke, christian.rohrdantz, daniel.keim, oliver.deussen}@uni-konstanz.de*
- *Andreas Stoffel, E-mail: andreas.stoffel.ext@siemens.com*

2 Related Work

2.1 General Approaches

Faced with the task of overviewing a document collection, the usage of technologies integrated in operating systems is a common solution. File browsers like Microsoft Windows Explorer or Apple Finder provide a thumbnail view of the first page of a document file. Setlur et al. [32] create document icons that include representative images from a web image database found by key text features of a file. Other thumbnail approaches discuss the use of 3D icons, which map each information on a side of a cube (Henry and Hudson [14]) while Lewis et al. [22] focus on distinctive icons as a graphical solution for the "lost in hyperspace" problem. Previewing technologies like Apple Cover Flow add the capability to browse through document pages in place. Cockburn et al. [9] show that representing all pages of a document in one view (Space-Filling Thumbnails) allows fast document navigation.

Visualizations for small devices aim at compact representations. Breuel et al. [5] propose to use and rearrange original text snippets from a text image to circumvent OCR parsing problems. Berkner et al. [4] extend this approach and create combined text-and-image thumbnails called SmartNails. The used images are scaled and cropped to automatically extracted regions of interest. How to

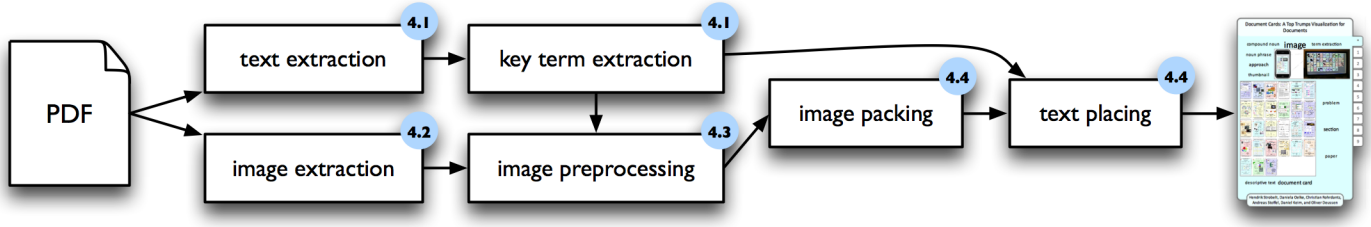

Fig. 2. The Document Card pipeline. Each step is further explained in the sections indicated by the number in the top right corner of each box.

find such regions is also described by Suh et al. [35]. Suh and Woodruff [36] introduced Enhanced Thumbnails which overlay and highlight extracted keywords on a scaled and saturation reduced version of a web page. The idea of creating thumbnails of PDF files is discussed by Sauer et al. [2]. They extract images from documents, sort them by their filesize, and arrange the "top few" of them on the frontpage. Berkner [3] discusses an approach of finding the best scale for a document page relating to its type of content. Lam et al. [21] introduced the concept of Summary Thumbnails, which represent webpages as thumbnail views, enhanced with shortened text fragments in larger font size. The main layout of the webpage remains (as well as the total line count). Erol et al. [10] use the audio capability of an handheld device to auto-generate a small film introducing a document. The film contains images, and the highly relevant terms are spoken.

Russell and Dieberger [27] describe how to automatically instantiate manually created Summary Design Patterns using texts and images.

Our approach combines images and key terms in a single Document Card. In contrast to other methods, we build a compact representation of a whole document that combines representative images with expressive key terms. These Document Cards can be used on large displays to browse large document collections as well as on handheld devices to get an overview of a single document.

2.2 Term Extraction

Approaches for keyword or key term extraction often originate from the information retrieval field like the prominent TFIDF method ([34], [28]). An extensive survey on Information Retrieval methods was published by Kageura and Umino [16]. But also in text mining research key term extraction methods play a role as pointed out by Feldman et al. [12]. Usually, a measure is defined to score terms with respect to a document or a document collection. A certain number of top scored terms according to the measure are then extracted as key terms.

Whereas most approaches require a suitable document corpus for comparison in order to extract key terms out of a single document, Matsuo and Ishizuka [24] describe a method that is able to extract key terms out of a single document without further resources. The approach is based on the co-occurrence of terms in sentences and the χ^2-measure to determine biased co-occurrence distributions in order to assess the importance of terms.

Our approach also uses an extension of the χ^2-measure to identify important key terms. However, we base our extraction method on the structure of the document. The rationale for this is explained in Section 4.1.

2.3 Image Extraction and Image Classification

Extracting images from a document format like PDF using standard tools is challenging. Chao and Fan [7] split PDF Documents into the components: images, vector images, and text. Maderlechner et al. [23] focus on finding figure captions and mapping them to images. Cohen et al. [18] mention to use a modified version of open source tools to extract images.

For image classification Chapelle et al. [8] suggest Support Vector Machines operating on image histograms. Moreno et al. [25] and Vasconcelos et al. [37] suggest to use the Kullback-Leibler divergence as distance measure between two histograms.

2.4 Layout

Placing a set of rectangular images optimally into a given rectangular canvas is known as rectangle packing. It is in the class of NP complete problems. From the wide range of algorithms which provide an approximative solution, three approaches are referenced here. A method from computer graphics uses efficient packing methods to create texture atlases [30]. Murata et al. [26] introduced the sequence pairs to transform the problem into a P-admissible solution space problem. The approach generates packings of high quality in reasonable time for offline use (like in VLSI design). Itoh et al. [15] have shown a fast, geometric algorithm for placing rectangles even in online time. The algorithm does not use global optimization, but produces packings of good quality. A survey of rectangle packing is given in Korf [17].

Seifert et al. [31] give an overview of recent approaches generating text clouds. Their approach describes an iterative algorithm for optimizing font sizes and string truncation to place text bounding boxes into given polygonal spaces. We adapt this approach in Section 4.4. Feinberg [11] provides a web service for creating text clouds. The used technique is not publicly described.

3 Design of the Document Cards

Summarization necessarily is a lossy compression and requires a decision of what can be preserved and what has to be excluded. Document Cards try to address this problem with special foci that are reflected in the following constraints and design decisions:

- *Document Cards are fixed size thumbnails that are self-explanatory.* Approaches like [5], [36], [21], and [10] preserve the main structure of a document on a fixed size view. But these approaches require interaction, like browsing or listening, to get a global insight into a document. In [3] the optimal scale for pages are calculated which breaks the constraint of a fixed size representation.
  As Document Cards shall also be applicable on small screen devices like handhelds or mobile phones it is an important feature that they provide meaningful global representations on a given limited space.
- *Document Cards represent the document's content as a mixture of images and important key terms.* Erol et al. [10] evaluated the most important parts of a document for the tasks of searching for it and understanding its content. Namely the top three are: title, figures, and abstract.
  Since we are aiming at a small representation we include the title (as top one feature), a filtered collection of figures, and we extract important keywords as an approximation for the content. Previous approaches, aiming at even smaller size representations, focus either on the semantic content (Semanticons [32]) or the contained images and image texts (SmartNails [4]), but not both. We present novel methods that carefully filter the most meaningful representatives of both categories and combine them in one view.
- *Document Cards should be discriminative and should have a high recognizability.* Summary Design Patterns [27] provide a uniform look on summaries of picture collections. In opposition

to that, Document Cards are designed to be easily distinguishable and recognizable. This is supported by the selection of meaningful images. Furthermore, the aspect ratio of images is preserved and the background of Document Cards is color-coded as described in Section 4.4.

4 Automatic Generation of Document Cards

In this section we describe the pipeline for creating Document Cards (see Figure 2). We show how to extract text from a PDF file and find the key terms (4.1). Further we present an image and caption extraction algorithm (4.2). We then discuss how images are scaled by their semantical weight and how we classify them (4.3). In Section 4.4, we show how to use the generated input to visualize a document as a Document Card.

4.1 Key Term Extraction

For each document we have to extract the key terms that describe the topics of the document. In the field of biomedical text mining the distribution of keywords in scientific publications has been examined several times. Shah et al. [33] searched for keywords in five standard sections and came to the conclusion that "information is unevenly distributed across the sections of the article, that is, different sections contain different kind of information". A study by Schuemie et al. [29] that also examined five standard sections had a similar outcome, which was that "30-40 % of the information mentioned in each section is unique to that section". Both studies come to the conclusion that abstracts, while having the highest keyword density, do not even nearly cover all the information (keywords) contained in a full-text scientific article.

Based on these findings we decided not to limit the term extraction to abstracts. Instead, we use full-text articles regarding the section boundaries also as topic boundaries. An author usually starts a new section, when writing about a different topic or sub topic. As a result, non-relevant terms will appear equally distributed over all sections of the document, while the important key terms will not. They have higher frequencies in the sections of their particular topics and a lower frequency in the others. Thus, the non-equally distributed terms are the key terms we are looking for.

At first we have to find the sections in the documents. We extract the text lines of the PDF files and train a machine learning algorithm on geometry, formatting, and text features to identify the headlines of the sections. In the same step we also distinguish between the continuous text of a section and other text information like headers, footers, tables, or captions. Scanning line by line through the document we discover sections by their headlines. A new section is started, when a top-level headline is hit. Afterwards, all the continuous text lines are added to the latest section until a new section is started.

4.1.1 Preprocessing and Candidate Filtering

The preprocessing comprises sentence splitting, part-of-speech tagging and noun phrase chunking with OpenNLP-Tools [1] and a base form reduction of words according to Kuhlen's algorithm [20].

Next, in the candidate filtering step we eliminate stopwords and noise. Verbs are also deleted, a decision that is based on the empirical observation that even verbs which have a characteristic distribution are of a rather general nature. For many papers the salient verbs are e.g. "work", "show" or "compute". Whereas approach-specific verbs mostly also appear in their nominalized form. For example, for the paper at hand it would be much more meaningful to get the terms "image extraction" or "term extraction" than the verb "extract".

4.1.2 Special Noun-Phrase Processing

Compound nouns, noun phrases consisting of at least two terms, have the highest potential to be very descriptive for a certain paper. Among the 130 index terms that the authors of the InfoVis 2008 publications manually assigned to their papers, 92 (about 70 %) correspond to compound nouns, which emphasizes their importance. This is because they often correspond to technical terms that are very specific and descriptive for a described approach.

At the same time we also consider sub phrases of larger noun phrases. The noun phrase "a term extraction algorithm" has several sub phrases that might be interesting. Our algorithm deletes leftmost articles like "a" and then builds every rightmost sub phrase, e.g. in this case "term extraction algorithm", "extraction algorithm" and "algorithm". In most cases by shortening the noun phrase in this particular way, the shorter representations are generalizations of the longer ones.

4.1.3 Term Scoring and Term Extraction

For the term scoring, the PDF file is scanned and the occurrence of every term is counted for each section separately. As a result, we get a vector for every term where each dimension corresponds to a section and each dimension's value is the number of occurrences of that term in the section. We keep only those terms that occur at least seven times in the document. All other terms are considered to be too infrequent to be key terms.

For each of the remaining vectors we calculate how strongly it deviates from an equal distribution using an extension of the χ^2-measure:

$$\chi^2_{sec}(t,D) = \sum_{s \in D} \begin{cases} \frac{\left(freq(t,s) - freq(t,D)\frac{size(s)}{size(D)}\right)^2}{freq(t,s)}, & \text{if } freq(t,s) > 0 \\ 0, & \text{else} \end{cases},$$

where D denotes the document, s the section and t the term. Accordingly, $freq(t,s)$ is the occurrence count (observed frequency) of term t in section s, $freq(t,D)$ the term's count in document D and $size(x)$ means the number of terms in a text unit x. The part $freq(t,D) \cdot size(s)/size(D)$ thus describes the expected frequency of a term t in a section s, if we assume equal distribution.

For every section, we calculate the squared deviation of the observed frequency from the expected frequency is summed up, after normalizing it by dividing it by the observed frequency. Usually in the χ^2-test the normalization is done by dividing by the expected frequency, which is changed here to avoid overestimating terms in very short sections. For example, a term that appears once within a section of 10 words in a paper of 1000 words. The summand for this term and this section would be $((1-1\cdot(10/1000))^2)/(1\cdot(10/1000))$ = 98 which is inappropriately high and would distort the overall result. With our normalization, the corresponding summand is only 0.98. The modification of the normalization still scores terms with strongly deviating distributions higher but without the undesired effect of potentially over-scoring terms that appear in very short sections. At the same time, sections where a term is not contained do not contribute to the term score. Hence, high scores are assigned to terms that not only have a skewed distribution but also are present in several sections. This guarantees that terms are preferred that not only appear in one section but ideally play a vital role in distinct parts of the document.

Despite their descriptive nature, compound nouns are usually not among the highest scored terms according to the described method. To improve the score of the compound nouns, we boost them by doubling their occurrence counts compared to normal terms.

After scoring the terms with our χ^2_{sec} scoring function, the top-k terms with the highest scores are extracted. If there are compound nouns in the top-k terms, which are contained by other compound nouns also present in the top-k, then the shorter ones are discarded and replaced by the terms with the next highest scores. For example, if the terms "extraction algorithm" and "algorithm" are present within the top-k terms, we delete the latter one keeping only the longest and thus most specific compound noun. The number k of terms to extract is determined by the available layout space in a DC.

The strength of our key term extraction approach is the corpus independence. Approaches like TFIDF that use a document corpus for comparison in order to score and extract terms are not applicable if there is no suitable comparison corpus available. In opposition to that, our method does not depend on additional data sources and can be applied only having a document itself. Furthermore, TFIDF prefers to extract terms that discriminate one document from the others and

in our approach we aim to extract descriptive terms for single papers that not necessarily have to be discriminating. Otherwise, topics that dominate a document corpus would not be extracted because of their lacking discrimination power. That means, if for example many papers within the InfoVis corpus tackle graph-related problems, we want the corresponding terminology to be extracted. This provides us with the valuable information that graph methods are common to many approaches — a fact that we otherwise might not be aware of.

4.1.4 Comparison to a Section-Based TFIDF Approach

As we possess section information, it would also be possible to apply a TFISF "term frequency inverse section frequency" approach to extract terms from a single document. In this case for a specific term we get several TFISF values per document, one for each section. Then, these values have to be mapped to a single term score for a term with respect to a document. One obvious option is to take the average TFISF value as term score:

$$\begin{aligned} avg-tfisf(t,D) &= \frac{1}{n}\cdot\sum_{s\in D}(tf(t,s))\cdot isf(t,D)) \\ &= \frac{isf(t,D)}{n}\cdot\sum_{s\in D}tf(t,s) \\ isf(t,D) &= log\left(\frac{n}{|\{s:t\in s\}|}+1\right) \end{aligned}$$

After the transformation of the equation we can single out three factors: $\sum_{s\in D} tf(t,s)$ is the sum of the frequencies the term t has in the different sections. It is possible to sum up either absolute or relative frequencies. As $1/n$ is just a constant factor that is the same for all terms the only other relevant parameter is the *isf* value, which takes into account in how many sections a term is present. A weak point of the formula is the binary nature of this *isf*: It makes no difference if a term is frequent or infrequent in a section as long as it occurs at least once. This drawback is a main disparity to our χ^2_{sec} approach. We applied both methods to test documents to review the implications of our idea on a practical scenario. We observed that the avg-TSISF approach tends to prefer terms that appear only in one section, which is not what we are aiming at.

4.2 Image Extraction

To extract the images and their associated captions we make use of the freely available tools ghostscript [13] and pdftohtml [19]. Ghostscript provides us with all the necessary information to render a version of a page that only contains the images. Pdftohtml is used to create an xml file describing the position, width, length, and strings of text boxes of each page. We combine the output of both tools to create a schematic map of images and text boxes as shown in Figure 3. In the map all image pixels are rendered in black, text boxes in red, and caption candidates in green. Here, the height of the text boxes is increased to receive continuous text clusters. A continuous text cluster is defined as a caption candidate if it starts with a caption indicating keyword such as "figure" or "table".
To extract the images, a scanline runs from top to bottom of the schematic map to find the image coordinates. An image is defined as the region between a first image pixel (black pixel) and a line containing a figure reference (green pixel) or a line which contains mostly standard text (mostly red pixels), in case we missed a caption text. For two-column documents we run three scanlines, one for the whole page width and one for each column.

Figure 3 shows examples for which the image extraction is difficult. On the left page the captions are written on top of the images and the images are visually not separated. The page on the right side contains an image that consists of four separate parts but that only has one single caption. The advantage of the approach is that it even performs well for such special cases.

Fig. 3. The schematic maps of two document pages.

4.3 Image Weight and Image Classification

Important images are slightly enlarged to make them more prominent. We consider an image as important if an important key term is found in its descriptive text. We use the concatenation of an image's caption and its referencing text as descriptive text. The referencing texts are sentences of the document that refer to the specific figure and are found by a regular expression. To map the importance of an image we define the following scaling function:

$$\begin{aligned} scale &= (1.0 + scale_{max}\cdot w_{max}) \\ size_{image} &= size_{image}\cdot scale, \end{aligned}$$

where w_{max} is set to the maximum weight of the key terms found in the descriptive text (as calculated in the term extraction step) and $scale_{max}$ is a constant factor that controls the influence of the key terms with respect to the size of the images. We experimentally set $scale_{max}$ to a value of 0.5. By this method we consider the combination of the original image size and a semantical size boost for later processing.

Next, we classify the images into one of the following categories:

- A *table (T)* is a set of facts systematically displayed. Its colors are mostly black and white.
- An *image of category (A)* is a diagram, a sketch, or a graph image which shows a concept or has explaining character. It uses a reduced number of colors.
- An *image of category (B)* is a photography or rendered image which shows a real world scenario or an expressive computer generated scenario. It is characterized by many colors, which have a rather complex distribution across the color space.

In the classification process each image is represented as an HSV color histogram with 8 values per channel and 8 values for grayscale, resulting in $8^3 + 8 = 520$ values per histogram. The values of the histogram are normalized by the total number of pixels and sorted in decreasing order. This allows us to compare different images with respect to their distribution of color usage across the color space instead of the specific colors they use. As recommended in [37], [25], and [8] we use Support Vector Machines (SVM) as the classification method (in our case the implementation that is provided by the LIBSVM library [6]) and the Kullback-Leibler divergence as distance function. We use a radial base function in the SVM and train it with 57 representative images from the IEEE Vis 2008 proceedings corpus. In the classification step, the most probable class label is assigned to an image. The usage of the class labels is described in Section 4.4.

4.4 Layout

In the previous sections we explained how to extract the images and key terms. The images are resized according to their semi-semantic

weight and assigned to one of the image classes. In this section we describe the transformation from these bag of terms and set of images to a compact view.

4.4.1 Image Handling

In each Document Card up to 4 pictures are shown. These are chosen as follows: First, images that have been classified as tables are omitted. The reason for this is that downscaled tables do not provide much information because their contained text in small font size is not readable. Only if no other image is available, a table is shown in the DC. Next, the remaining images are sorted with respect to their size (the size is influenced by their semi-semantic weight as explained in 4.3). We take the first four images of the list to display them in a Document Card. During this process, we check if the following two constraints are fulfilled: a) We want to display at least one image from each category. Thus, if there is no image of category A (or B) included in the list, the last image in the list is discarded and substituted with the the largest image of category A (B, respectively). b) If the area of an image in the list is smaller than 25% of the area of the largest image, the image is discarded. This is done to avoid the insertion of too small images.

After filtering, the images are packed into the DC canvas. Packing of image bounding boxes to fit optimally in size to a given aspect ratio is an NP complete problem. Therefore, a good approximation is needed, that provides a fast solution with good results. Itoh et al. [15] have presented such an algorithm which we adopted and extended. They suggest to use a penalty function for each image insertion which respects the bounding box increase and the difference from the aimed bounding box aspect ratio. Our extension takes the original position of an image on a page into account for defining a new penalty function argument. That means, that images appearing in the upper right of the original page tend to appear up right in the summary visualization. This optimizes animation of the interaction part. After arranging the bounding boxes, the calculated layout is scaled to fit into the DC canvas.

The images are positioned iteratively on the Document Card according to the coordinates that are given by the packing algorithm. At the same time we collect information about the free areas of the canvas that the key terms will be placed in later. This is done as follows: For each insertion of an image the surrounding free space rectangle is split into up to 4 new rectangles located on the top, bottom, left, and right side. In Figure 4 the procedure is illustrated for an insertion of the first and second rectangle. After inserting the first image at its calculated position, the DC canvas is split into a left, right, top, and bottom section (left side of Figure 4). The second image is placed under the first one in this example. It splits the free space rectangle at the bottom into three new sections: left, right, and bottom (right side of Figure 4). By splitting the canvas with horizontal lines we support the creation of free space rectangles with mostly a $width/height$ ratio larger than one. This is important for placing text items in these free spaces since they have a $width/height$ ratio much larger than one. The following algorithm details the process:

a list L_i of images with calculated positions;
a list L_r of free space rectangles;
initialize L_r with the DC canvas bounding box;
for all i in L_i **do**
  for all r in L_r that intersect i **do**
    split r into r_T, r_B, r_L, and r_R;
    add all r_X to L_r;
    remove r from L_r;
  end for
end for

4.4.2 Text Handling

The term extraction (4.1) outputs a list of terms with associated weight. To place the terms into the canvas, we extend the idea given in [31]. In order to avoid term overcrowding and to guarantee good readability the number of terms that is shown in a DC depends on the size of the available free area after the image positioning step. The number of terms (n) that is displayed on a DC is determined as follows:

$$n = \left\lfloor \kappa \cdot \left(\frac{A_{DC} - A_{Im^*}}{A_{DC}} \right) \right\rfloor$$

where A_{DC} is the total area of the Document Card, A_{Im^*} is the cumulated area of all placed images, and κ is a constant of maximal terms that should fill an empty DC.

We use the font size to indicate the relative term weight. Using and mapping different font sizes is a critical part in the layout process because text is less scaleable than graphics. To ensure readability, the variation of the font size of the different key terms has to be limited to a small range. The font size s_i for a key term i is calculated as follows:

$$\begin{aligned} s_i &= s_{max} \cdot scale_i \\ scale_i &= \frac{s_{min}}{s_{max}} + \beta \cdot \left(\frac{w_i - w_{min}}{1.0 - w_{min}} \right), \end{aligned}$$

where s_{max} is the maximum font size, s_{min} is the minimal font size, w_{min} is the minimal term weight for a document, and w_i is the term weight for term i. The value of β can be varied between $0 \leq \beta \leq 1 - \frac{s_{min}}{s_{max}}$ depending on the available free space on the DC.

To position the text boxes in the canvas we first sort the free space rectangles that have been collected in the image extraction step by decreasing size. The list of key terms is sorted by term weight. Iterating over the list of terms and rectangles a term is positioned in the center of the first rectangle that is large enough to host it. Afterwards the rectangle is split like in the image extraction step to detect the remaining free space in this rectangle. If there are remaining terms after the procedure that cannot be positioned anymore, β is decreased and the process is repeated. The algorithm terminates when all terms are positioned or $\beta < 0$.

By using this algorithm we try to position the important terms in the middle of free space areas. This will support stability in future advanced (semantic) zooming approaches.

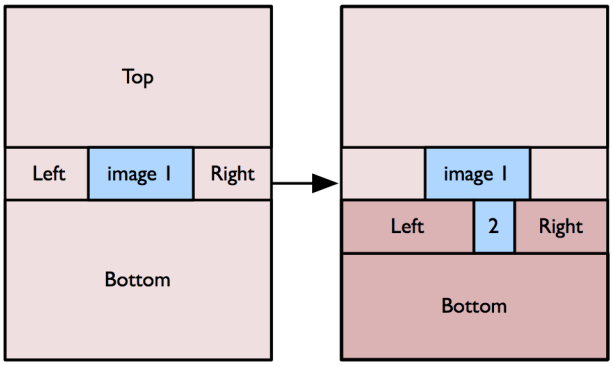

Fig. 4. The split algorithm used for finding empty space rectangles: After insertion of image 1 the canvas is split into 4 regions. The bottom region is further split into 3 new regions on insertion of figure 2.

4.4.3 Finishing

After positioning the images and keyterms, the Document Card is enriched with the document title and the documents author names. We further add a page number list at the right side of each card to show the overall size of a document and to allow navigation as explained in the Application Section (5). For better discrimination we color code the background of each DC in the following way: From all images positioned in one DC canvas we evaluate the most frequent color value (H value in HSV color model). We use this color value less saturated as background color.

5 Application

We applied the DC approach to the InfoVis proceedings of 2008. Figure 6 shows the corpus as a matrix of Document Cards. The tool provides the following interaction features for a Document Card (DC):

- Hovering over the non-image space in a DC shows the extracted abstract of the document as tooltip.
- Hovering over an image displays the image's caption as tooltip.
- Clicking on a page number (right side of a DC) starts a transition to the full page (see for example DC 3 in Figure 6 that has been switched to page mode and shows page 2).
- Clicking on an image starts a transition to the page containing the image.
- Clicking on a term highlights the term in the overview and in all tooltips for this document. Additionally, all images containing this term in their descriptive text are highlighted. The term density is shown in the page indicator on the right side of a Document Card. The higher the density of the term on a page the less transparent is the corresponding tab. (This technique is shown in DC 12 of Figure 6 in which the term "tree diagram" has been selected).

The hovering approaches provide readability of the caption and abstract text even if the DC is in small scale. To show the connection between images and terms we introduced the idea of highlighting on term clicking. Our supplementary video illustrates these interaction ideas.

5.1 Analyzing the InfoVis 2008 proceedings

Figure 6 gives a quick overview of the InfoVis 2008 paper collection.[1] Skimming over the DCs it is easily perceivable that many graph-based techniques have been accepted to last year's conference (e.g. DC 0, 4, 7, 11, 13, 17, 22, 24). Such a first impression of the content of the collection is usually based on the images that are depicted. As InfoVis is a visualization conference images are naturally very expressive with respect to what a paper is all about. However, if only the images were given it would be hard to tell in which area of graph-related approaches a paper contributed. The title of the paper and the automatically extracted terms help to clarify this. They reveal for example that DC 0, 4, 7, and 17 are all papers that deal with graph layout algorithms. DC 13 represents a paper that proposes a visualization approach for large power-law graphs. Omitting unimportant information is important here which is also reflected in the terms of its DC (e.g. "simplification method, edge filter, ..."). On the other hand the papers of DC 22 and 24 conducted user studies related to graph layout and the impact of data transformation techniques respectively. Consequently, terms such as "experiment, study, human observer, and anova test" appear on their DCs. The above examples show that both the graphical and the textual information of a paper are important to convey its content.
Finally, DC 14 and 15 are interesting because they do not contain any visualizations at all (but only schematic diagrams). Although this might seem strange at first for a visualization conference there is a simple explanation: Both papers contribute with theoretical work in the context of Information Visualization instead of presenting novel visualization techniques.

5.2 Applicability on Large and Small Devices

Document Cards are suitable for both the visualization of single documents on small devices and to provide an overview of large document collections on larger devices. Figure 5 shows an example for both scenarios. The left side of the figure depicts a mockup of a Document Card on a mobile device. On the right side you can see the whole collection of the IEEE Vis 2008 proceedings (consisting of 46 papers) on a 56 inch display with high resolution.

[1] Please note that one of the papers was not automatically parsable and therefore does not show up in the DC matrix.

6 Conclusion

Document Cards provide a meaningful and representative small-scale overview of a document that is applicable for a broad range of document types and display sizes. We present a pipeline for the automatic creation of Document Cards with contributions in several subtasks. The main contributions of our approach are novel methods to automatically extract and select the most expressive images and the most descriptive key terms out of a document.
The top key terms are extracted by an advanced text mining approach that combines automatic document structure extraction with an extension of the χ^2-measure. Terms with a characteristic intra-document distribution are extracted which makes the approach independent from further data resources as e.g. corpora for comparison. In addition, meaningful compound nouns get a higher weight in our approach as they are generally very descriptive and document-specific.
During image extraction, we combine a novel semi-semantic image weighting with an image classification approach. An image's weight depends on the presence of the previously extracted key terms within its descriptive text. Both images and their descriptive texts (captions and reference sentences) are extracted in a fully automatic way. For this purpose we developed a new method that is capable to solve even problematic cases, e.g. when captions are printed on images. Finally, an image classification is applied with the purpose to capture images that are still meaningful in a small-scale picture. This implies that e.g. we do not consider tables or thin-lined graphs. Furthermore, the classification allows us to prefer the insertion of at least one representative of each image class if the space is too limited to insert all potentially useful images.
In application Document Cards support tasks like browsing, recognition, and acquiring an overview of whole sets of documents. Therefore, they were enhanced with a wide range of interaction features. In a preliminary user study the users stated to be more efficient and have more fun browsing paper collections with document cards than with a PDF reader.

In future work, we aim to include Document Cards in larger visualizations like clusterings, author networks, or citation networks. These techniques can introduce better interaction approaches for full collections. We want to investigate if the embedding of image's references in the document structure can improve the selection process. Furthermore, the creation of an enhanced semantic zooming approach is planned. Depending on the current space availability more or less content will be provided. As part of the interaction improvement and semantic zooming we will investigate cluster-wide text features for each document and the cluster itself and therefore allow a view on a document in its semantical surrounding. Key terms are scheduled to be extracted in a hierarchical manner for clusters and documents possibly combining our approach with a TFIDF-like technique. In addition we will apply image enhancement. Some images with sparse or high frequent content could be enhanced by finding regions of interest and stress these regions by providing more abstraction or highlighting their main structure. As part of image enhancement we will introduce approaches for external image acquisition for text only documents. We will test different layout approaches that amplify the relation between images and texts.

Acknowledgments

The authors gratefully thank Josua Krause for reducing the workload of creating a good prototype. This work has partly been funded by the German Research Society (DFG) under the grant GK-1042, Explorative Analysis and Visualization of Large Information Spaces, Konstanz, and by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

References

[1] J. Baldridge. The opennlp project. `http://opennlp.sourceforge.net/`, 2009.

[2] D. Bauer, P. Fastrez, and J. Hollan. Spatial Tools for Managing Personal Information Collections. *Proc. of Hawaii Int. Conf. on System Sciences*, 4:104b, 2005.

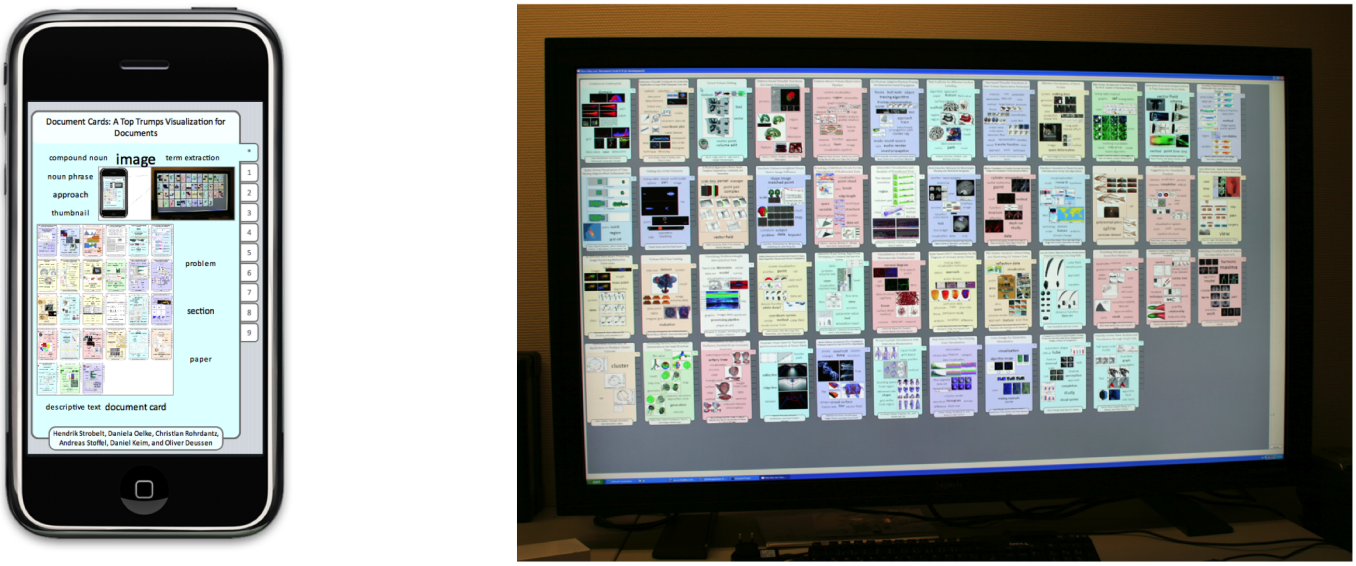

Fig. 5. Applications for different screen sizes: On the left side a mockup of a Document Card on a mobile device. On the right side the IEEE Vis 2008 proceedings corpus displayed on a 56 inch display.

[3] K. Berkner. How small should a document thumbnail be? *Proc. of SPIE*, 6076:127–138, 2006.

[4] K. Berkner, E. Schwartz, and C. Marle. Smartnails – Display- and Image Dependent Thumbnails. *Proc. of SPIE*, 5296:54–65, 2003.

[5] T. Breuel, W. Janssen, K. Popat, and H. Baird. Paper to PDA. *Proc. 16th ICPR*, 4:476–479, 2002.

[6] C.-C. Chang and C.-J. Lin. LIBSVM – A Library for Support Vector Machines. `http://www.csie.ntu.edu.tw/~cjlin/`, 2009.

[7] H. Chao and J. Fan. Layout and Content Extraction for PDF Documents. *Document Analysis Systems VI*, pages 213–224, 2004.

[8] O. Chapelle, P. Haffner, and V. Vapnik. Support Vector Machines for Histogram-Based Image Classification. *IEEE Trans. on Neural Networks*, 10(5):1055–1064, 1999.

[9] A. Cockburn, C. Gutwin, and J. Alexander. Faster Document Navigation with Space-Filling Thumbnails. *Proc. of CHI*, pages 1–10, 2006.

[10] B. Erol, K. Berkner, and S. Joshi. Multimedia Thumbnails for Documents. *Proc. of the 14th ACM Intern. Conf. on Multimedia*, pages 231–240, 2006.

[11] J. Feinberg. Wordle - Beautiful Word Clouds. `http://wordle.net/`, 2009.

[12] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. Text Mining at the Term Level. In *Proc. of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98)*, pages 65–73, 1998.

[13] Ghostscript. `http://www.ghostscript.com/`, 2009.

[14] T. Henry and S. Hudson. Multidimensional Icons. *ACM Trans. on Graphics (TOG)*, 9(1):133–137, 1990.

[15] T. Itoh, Y. Yamaguchi, Y. Ikehata, and Y. Kajinaga. Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 10(3):302–313, 2004.

[16] K. Kageura and B. Umino. Methods of Automatic Term Recognition: A Review. *Terminology*, 3(2):259ff, 1996.

[17] R. Korf. Optimal Rectangle Packing: Initial Results. *Proc. of the 13th Intern. Conf. on Automated Planning and Scheduling (ICAPS03)*, pages 287–295, 2003.

[18] Z. Kou, W. Cohen, and R. Murphy. Extracting Information from Text and Images for Location Proteomics. *Proc. 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD03)*, pages 2–9, 2003.

[19] M. Kruk. PDF to HTML conversion tool. `http://sourceforge.net/projects/pdftohtml/`, 2009.

[20] R. Kuhlen. *Experimentelle Morphologie in der Informationswissenschaft.* Verlag Dokumentation, 1977.

[21] H. Lam and P. Baudisch. Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. *Proc. of CHI*, pages 681–690, 2005.

[22] J. Lewis, R. Rosenholtz, N. Fong, and U. Neumann. VisualIDs: Automatic Distinctive Icons for Desktop Interfaces. *ACM Trans. on Graphics (TOG)*, pages 416–423, 2004.

[23] G. Maderlechner, J. Panyr, and P. Suda. Finding Captions in PDF-Documents for Semantic Annotations of Images. *LNCS: Structural, Syntactic, and Statistical Pattern Recognition*, 4109:422–430, Jan 2006.

[24] Y. Matsuo and M. Ishizuka. Keyword Extraction From A Single Document Using Word Co-Occurrence Statistical Information. *Intern. Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.

[25] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. In *Advances in Neural Information Processing Systems 16*, 2004.

[26] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. Rectangle-Packing-Based Module Placement. *Proc. of Intern. Conf. on Computer-Aided Design (ICCAD)*, pages 472–479, 1995.

[27] D. Russell, A. Dieberger, I. Center, and C. S. Jose. Synthesizing Evocative Imagery Through Design Patterns. *Proc. of Hawaii Int. Conf. on System Sciences*, page 4pp, 2003.

[28] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620, 1975.

[29] M. J. Schuemie, M. Weeber, B. J. A. Schijvenaars, E. M. Van Mulligen, C. C. Van Der Eijk, R. Jelier, B. Mons, and J. A. Kors. Distribution of Information in Biomedical Abstracts and Full-Text Publications. *Bioinformatics*, 20(16):2597–2604, 2004.

[30] J. Scott. Packing Lightmaps. `http://www.blackpawn.com/texts/lightmaps/`, 2009.

[31] C. Seifert, B. Kump, W. Kienreich, G. Granitzer, and M. Granitzer. On the Beauty and Usability of Tag Clouds. *Proc. of the 12th Intern. Conf. on Information Visualization (IV)*, pages 17 – 25, Jun 2008.

[32] V. Setlur, C. Albrecht-Buehler, and A. A. Gooch. Semanticons: Visual Metaphors as File Icons. *Computer Graphics Forum (Eurographics)*, 24(3):647–656, 2005.

[33] P. K. Shah, C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Information Extraction from Full Text Scientific Articles: Where Are the Keywords? *BMC Bioinformatics*, 4(1), 2003.

[34] K. Spaerck-Jones. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28:11–21, 1972.

[35] B. Suh, H. Ling, B. Bederson, and D. Jacobs. Automatic Thumbnail Cropping and Its Effectiveness. *Proc. of the 16th ACM Symp. on User Interface Software and Technology (UIST)*, pages 95–104, 2003.

[36] B. Suh, A. Woodruff, R. Rosenholtz, and A. Glass. Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces. *Proc. of CHI*, pages 251–258, 2002.

[37] N. Vasconcelos. On the Efficient Evaluation of Probabilistic Similarity Functions for Image Retrieval. *IEEE Trans. on Information Theory*, 50(7):1482–1496, 2004.

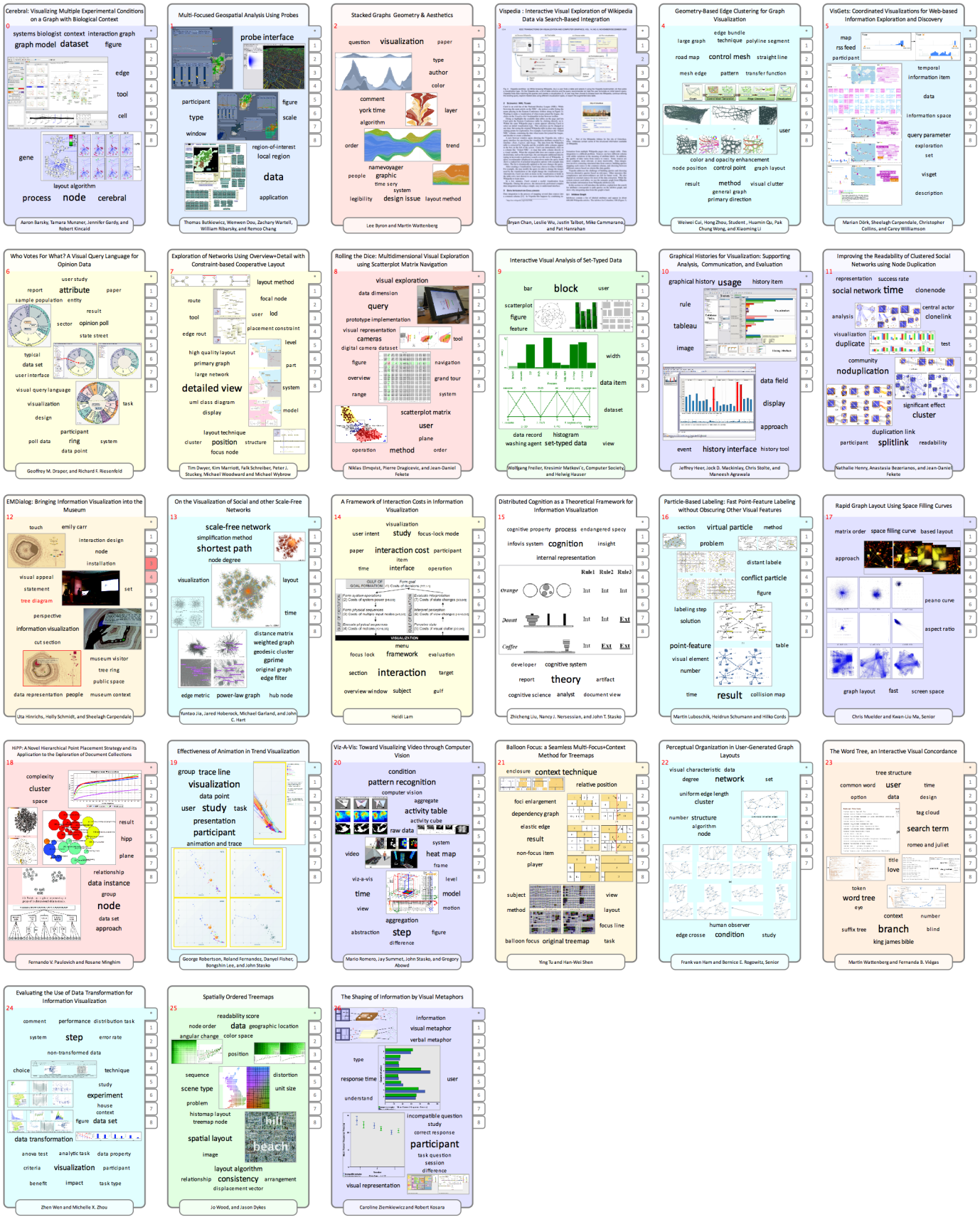

Fig. 6. The IEEE InfoVis 2008 proceedings corpus represented by a matrix of Document Cards (DC). DC 3 has been switched to the page view on page 2. In DC 12 the term “tree diagram” has been clicked. This highlights the image where the term occurs in its caption (on the bottom of DC 12). The frequency of the term on each page is shown on the right side of the DC (the more red, the higher the frequency).