TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERS THESIS

Functional Semantic Analysis of Web Pages on the Visual Layer

The REDEVILA system

written by

Bernhard Pollak

at the Institute of

Information Systems
Database and Artificial Intelligence Group
Vienna University of Technology

supervised by

o.Univ.Prof. Dipl.-Ing. Dr.techn. Georg Gottlob

and

Dipl.-Ing. Dr.techn. Wolfgang Gatterbauer

Vienna, December 2007 v1.2, 2008-02-04

Abstract

This masters thesis is motivated by the fact that data records on web pages are structured not only by word content but also by an implied visual hierarchy. A model of this visual hierarchy can greatly support automatic information extraction approaches become more domain independent and robust against variations of HTML syntax changes because the representation of information on the visual layer has to remain rather constant so as to remain understandable by humans. We refer to this visual layer as *functional level* which expresses the functional support for humans when structuring information visually.

This masters thesis first gives a thorough literature overview on (visual) document analysis and then presents such a functional level record detection system named REDEVILA (REcord DEtection on the VIsual LAyer). The approach works by superimposing a multi-topological grid onto the visual layer of web pages serving as an efficient spatial reasoning data structure for detecting the functional semantics between data items or data records. The system is principally domain independent as long as the layout hierarchy provided by the web page mainly depends on general topological and geometrical characteristics such as font size, distance and indention and not on color properties or word semantics. We further propose a novel diagonal ordering scheme to obtain a more "natural" or human-intuitive ordering and demonstrate the concept and problems of the visual based detection of single records. For the experimental evaluation we selected web pages from four different domains (blogs, search results, personal homepages and online newspapers) to show the basic domain independence of our system. Experiments were performed on 85 web pages and achieved a fair overall performance. We conclude that, while in its early stages, the visual approach has the potential to significantly improve the performance and robustness of traditional wrapper systems to induce a higher level of generalization and represent a next step towards generic web wrapping.

Acknowledgements

I wish to thank my wife Claudia for encouraging me to restart my computer science studies, for her continuous support throughout the finishing of my master's thesis and her help through discussions, extensive corrections and literature research (thank you so much for helping me with the bibtex entries). My mother Michèle supported me over all the years and I thank her for her belief in my success and her patience with me.

My supervisor Wolfgang Gatterbauer gave me the chance to get a deep insight into scientific work and I wish to thank him for my very first paper we wrote together and for his supervision during the writing of this master's thesis. I am honored that Prof. Georg Gottlob was so nice to be the main examiner of the master's thesis and I wish to thank him for his overall support and the freedom to define the topic of this thesis by myself.

Furthermore I wish to thank Prof. Georg Nagy at the Rensselaer Polytechnic Institute for sending me his two papers "Document Analysis with an Expert System" and "A Prototype Document Image Analysis System for Technical Journals" per airmail. To my great surprise he sent me not just copies but original papers. Thank you so much and as I said, I will frame the papers above my desk.

I would also like to thank Liu Wei at the Renmin University of China for answering me some questions regarding the VIPS algorithm and sending me a presentation with some additional information. Thank you very much.

Contents

1 Introduction

1.1 Background and Motivation

The term *information extraction* (IE) can have a wide variety of meanings depending on the specific context or domain in which it is used. In general the notion of information extraction includes all techniques for converting data or information which is targeted for humans to a machine understandable form, especially when this data is only available in an unstructured format and without any special meta tags. Given that the largest amount of information in a single source is available today on the Web in such unstructured or semi-structured formats, it is clear that the systematic extraction of information out of web pages has an enormous economic potential to support all kinds of data mining and data analysis efforts and to help people to canalize and make sense of the mass of information they are all facing today.

At the time of writing, the most common and also commercially used methods for information extraction from web pages are based directly on the html source, particularly the html tag structure. The consequence of this focus on the html source is a strong dependence of the used algorithms on the general template of the web pages which are selected as the prototype class. However, as there are so many different methods for presenting the same content in general and because of the evolvement of html, cascading style sheets (CSS) and JavaScript, web page designers use a wide variety of possible implementations resulting in the fact that learned wrappers operating on the html tag level cannot be generalized to other templates that easily even if the target web pages are from the same domain. While life would be easier if web pages are written reflecting only the structure and not the layout as originally intended, this is not common in today's standard of web page design. Figure 1.1 illustrates this idea: when analyzing two different web pages a classic tag based wrapper approach has basically to apply two different templates whereas the visual approach needs only one single set of visual rules based on the visual layer. Of course the resulting functional semantic could only express record items and hierarchical structure whereas a wrapper approach extracts the full semantic content. The repeated structure detection on tag level generalizes also very well but we could suppose a kind of bijective relationship between repeated tag structures and repeated visual entities and conclude that this tag structures are a consequence of the intended visual similarity and therefore expressing visual semantics.

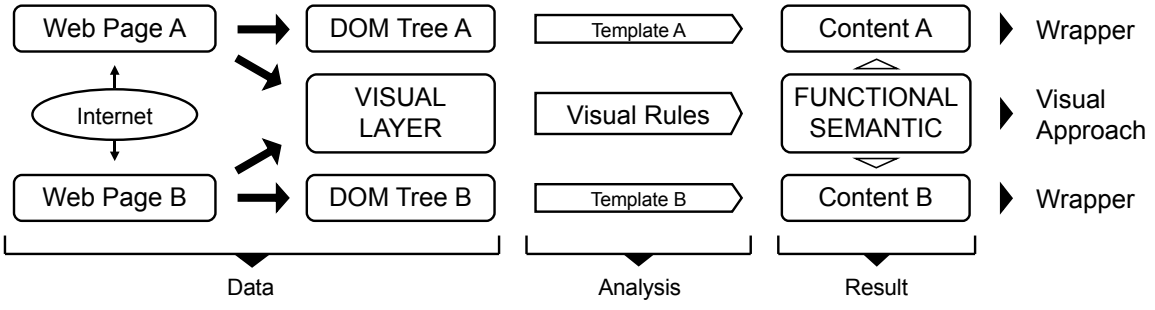


Figure 1.1: General difference between a traditional wrapper and a visual approach

When faced with visually presented information humans do not only analyze the semantic meaning of the content but also the geometric and layout related features. This analysis is also applied even without reading the text which leads to the conclusion that there exists an additional layer between the geometric features and the semantic labeling of words. We will refer to this additional layer as *functional level* ([45]) which expresses the functional support for humans when structuring information visually before even looking at the specific meaning of words. This *visual functional level* includes the visual record separation and visual determined (sub)structures and hierarchies. But there is still a grey zone because of the semantic expectations which allow humans to apply much more "wordless" semantic estimation as only the record separation and the hierarchy (e.g. the location of an address block inside a letter or the headline of a newspaper) causing an overlapping between the word level semantics and the functional level (see also figure 1.2).

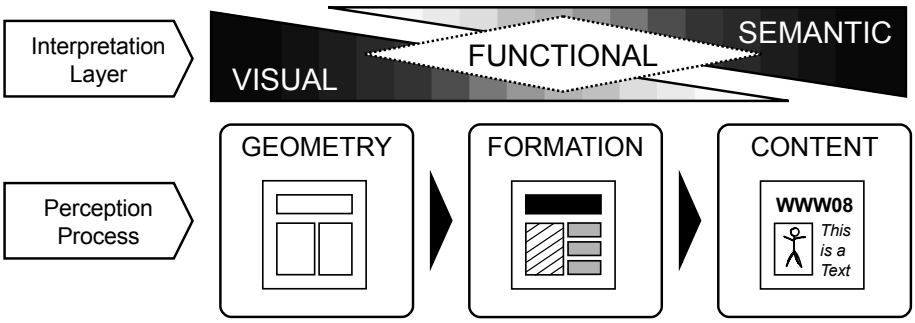


Figure 1.2: Correlation between the visual, the functional and the semantic layer during the perception process

The primary goal of this master's thesis is the detection and interpretation of such domain-independent visual functional semantics from web pages and, as a result, providing a potential additional semantic level for traditional html tag based attempts. Therefore it is not a stand-alone information extraction system but a system that assists existing non-visual approaches.

One side effect of this approach is that the visual insights could be easier transfered to other types of documents. For example, the methods could be used to analyze PDF documents if the positional and typographical information is provided. However, special considerations regarding the difference between scanned documents and web pages (e.g. the web page navigation or advertising).

In contrast to our previous work in our group [58, 59, 57] which focused on table data structures, the system presented in this work is targeted at substructured lists. A perfect distinction between tables, lists and substructured text is not possible but for our purpose, we will define substructured lists as follows:

Definition 1.1 (Substructured List)**.** *A two-dimensional substructured list is a series of similar data items and can be either one-dimensional or two-dimensional; in contrast to tables, no specific semantic relationships in between individual list items is implied except for a hierarchical structure and a possible ordering of the items. In turn, each data item itself is not an atomic entity, but is rather composed of nested hierarchical and semantic relations.*

Figure 1.3 shows a possible classification of spatially structured data with tables and substructures. Our VENTEX table extraction approach [59] focuses on the right table area (tables) and the REDEVILA system analyses list structures on the left side.

One remaining problem for the visual web page analysis research is the right balance between tag based approaches and visual based ones. For example, the VIPS algorithm [27, 28] contains a tag based part for the weighting and a visual part during the projection phase. Similarly, the approach

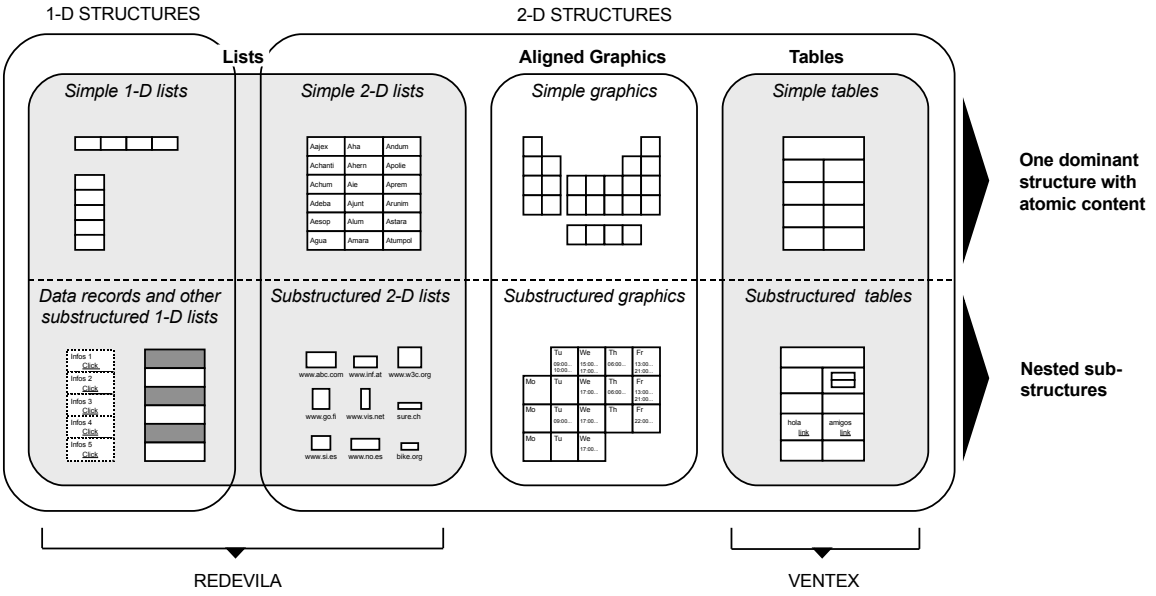


Figure 1.3: Spatially structured data with the VENTEX and the REDEVILA approach (from [59])

by Zou et.al. [153] is also based on a visual/tag combination but no clear distinction is made. For a definite classification we will introduce the concept of *Absolute Positioning Safe* (see also figure 1.4).

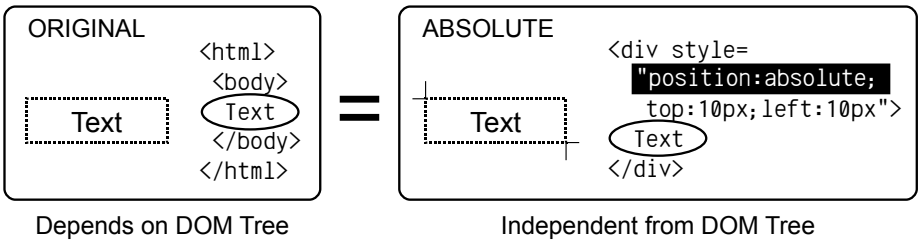


Figure 1.4: *Absolute Positioning Safe* (APS) concept

Definition 1.2 (Absolute Positioning Safe)**.** *An Absolute Positioning Safe (APS) extraction algorithm works transparently on whether the web page it is presented with is the original one, or an visually equivalent page where all information is placed with the help of absolutely positioned DIV elements.*

If a web page is presented to the algorithm twice first with the original HTML code and second with random absolute positioned DIVs reflecting the same web page in a visual exact way, the result of an APS algorithm is the same because it depends not on specific positions or a specific sequence inside the DOM tree.

1.2 Contributions

The main contributions of this master's thesis can be summarized as follows:

Comprehensive literature survey: We give a comprehensive analysis of approaches in the literature of general document analysis which are related to our task of visual web page analysis including an in-depth presentation of selected visual algorithms and methods.

REDEVILA system: We develop a visual functional semantic analyzer called *REcord DEtection on the VIsual LAyer* (REDEVILA) which detects records in a visual way by operating on the web browsers DOM tree building a visual representation of a web page.

The multi-topological grid: We develop an efficient grid structure that superimposes logical entities onto the content of the web page for the visual rule based spatial reasoning and the extraction of the visual functional semantic out of a web page (see section 5.6)

Diagonal ordering: We propose a diagonal ordering algorithm based on the maximal page width and the actual segment width which allows a more natural ordering than simple left/top or top/left ordering but remains sortable by pairwise comparison (see section section 5.7)

Visual single record detection: Our visual bottom-up approach works by detecting record header candidates directly by the reasoning algorithm and is therefore conceptually able to detect not only multiple records but also single records (see section 5.9 for examples and related problems). In contrast, recent visual based approaches need a minimum of two records (e.g. [94]).

1.3 Outline

Chapter 2 gives a comprehensive overview of related work in the area of general document and web page analysis. We try to give a rough classification on subchapter level based on visual properties.

Chapter 3 details some of the algorithms mentioned in chapter 2 as well as a few new additions. The focus lies on visual related methods like the X-Y tree, projection profiles or the VIPS segmentation algorithm and includes explaining figures and graphics.

Chapter 4 investigates the problems of saving web pages locally in a visual exact way and presents the WebPageDump solution which is later used for the experiments. Parts of the work of this chapter have been previously published in our SOFSEM 2007 (33rd International Conference on Current Trends in Theory and Practice of Computer Science) student research paper [112] with some graphics taken from the presentation held at the conference.

Chapter 5 starts with the concept of the *visual functional semantic* described by Doermann et.al. [45] and is followed by a detailed description of our visual functional level analyzer REDEVILA (REcord DEtection on the VIsual LAyer). Section 5.2 outlines the user interface and sections 5.3, 5.4 and 5.5 describes the box identification, the segmentation and the classification phase of the system. The next three sections 5.6, 5.7 and 5.8 cover the multitopological grid, the diagonal ordering approach and the hierarchy analysis. The last section 5.9 gives a thorough evaluation of the system and shows the potentials and problems for the separation of records based only on visual related reasoning.

2 Literature Survey

The segmentation and analysis of document structure has a long history, especially when measured inside the aera of computer science, where the development is (or seems) probably more faster than in traditional sciences and the roots of "computer archeology" go back only 100 years. With the raise of computers utilizing graphical features and scanner technologies, the need to manage and analyse printed material came up. This led to first attempts of optical character recognition and segmentation. During these times, researchers believed that computers would solve the problems with printed documents by the vision of the "paperless office". It is an irony of history that, with the increasing complexity of the visual design created and used by computers, similar problems with the visually oriented, but digital documents arise (and beside this, we are still using much physical paper).

In the following literature survey, the summaries inside the sub sections are ordered according to the publication year. The focus of the general document analysis part are the historical developments followed by a hopefully nearly complete overview about the web page analysis. As an importance guide we based the literature research on some general surveys and PhD Thesis ([63], [135], [8], [97], [73], [34], [30]). The survey of Chang et.al. [34] provides a very good and detailed overview about web information extraction systems, however misses a few important ones like the Lixto system. For historical related literature Nadler [104] presented a large but less detailed survey about document segmentation and coding going back to 1972.

2.1 General Document Analysis

2.1.1 Geometry and Topology

One of the earlier attempts is described by Wahl, Abele and Scherl for the detection of long vertical and horizontal lines. Wong and Casey together with Wahl presented an improved version in [145] that combines a horizontal and vertical 2D bitmap through a logical AND forming a final segmentation. The focus of the described segmentation algorithm was to distinguish picture and text regions for applying a text recognition.

Nagy and Seth [105] presented a tree data structure which they called X-Y Tree, where each node corresponds to a rectangle and each successor of a node is obtained by strictly alternating horizontal X-cuts and vertical Y-cuts with the first root node set to either horizontal or vertical. The decision for the cutting position is based on page grammars using pixel vectors across the width or height of the currently examined rectangle. They described the page grammar as "a generic tool for directing the search for cuts using a 'knowledge base' which stores specific information about the rules for cutting at different nodes of the derivation tree" [105].

Nagy, Seth and Stoddard [106] showed a labeling approach based on a rule set using their X-Y tree mentioned before. A rule is a Boolean combination of one or more conditions which are evaluated for a block B from the X-Y tree. They described the labeling as "a function from the set of blocks to the set of all possible labels". The label association with a block is satisfied "if and only if all the applicable rules evaluate to be true" [106].

Wang and Srihari [140] presented a newspaper classification system based on recursive X-Y cuts with projection profiles (the same techniques like [105] above). For the text detection they used the frequencies of black/white pixel runs represented by a Black-White-Black Combination Run Length Matrix (EWB Matrix) from which three features are derived.

Akiyama and Masuda [2] (an English version was published in [3]) demonstrated a segmentation method based on horizontal and vertical projections profiles. An interesting aspect of their work is that one of the features of their approach is the detection of line orientation due to the fact that in Japanese there exists the possibility of writing in horizontal and vertical direction. For the syntactic labeling based on headline, text lines, figures and tables the authors used some domain knowledge. The segmentation is also supported by the detection of stroke densities and the text size is done with projection profiles as well.

Tsuji [137] proposed an advanced document image analysis method using relative relations like the inclusion, left-right and top-bottom. The segmentation is carried out using projection profiles. For each sub block a so called F ratio is calculated that is defined by variance, mean positions and black-pixel frequency. The classification is obtained by applying either separation or reconstruction operations until the class of a block could be assigned. For this task inspection rules are formulated based on the block features. The nodes of the resulting syntactic tree are the element blocks and the other nodes represent the imaginary blocks.

Baird, Jones and Fortune [10] created many black rectangles out of the text at the char level and included a skew detection. The implied maximum white rectangles are calculated using an efficient enumeration algorithm. A maximum white rectangle is defined containing "only white and cannot be further expanding while staying all white. Clearly a maximum white rectangle touches black or the edge of the image on each of its four sides." [10]. Afterwards a so called cover set representing a subset of the maximum white rectangles is selected by applying a partial ordering on the rectangles. This ordering is driven by domain knowledge about the Manhatten layout and, for example, favours rectangles with high aspect ratios because they are likely to be column separators. Rectangles which are too small or thin are removed. For this the authors introduced "a shape score equal to the product of its area and the logarithm of its truncated aspect ratio" [10] that forms the sorting criterion and is therefore described as shape directed segmentation.

Spitz [132] proposed a document layout analysis based on styles where the user is able to interactively label the document. The style representation was done with SGML and is one of the earliest approaches utilizing a markup language. The approach was later modified by using XML as the base ([133] and [134]).

Pavlidis and Zhou [111] described a skew tolerant segmentation algorithm based on RSLA and white stream detection. The skew is detected after the segmentation and uses the centres of the column intervals instead of applying the algorithm directly to pixel line level resulting in fewer coordinates and easier computation. Following this steps a text vs. illustrations classification is done by signal cross-correlation.

Dengel et. al. [43] showed the ODA (Paper Interface to the Office Document Architecture) system with RSLA segmentation based on a layout and a logical tree structure for providing complementary views on a document. The system produces output which is compatible with the ODA standard [138].

Hirayama [66] presented an algorithm based on layout analysis using the border lines of text blocks to handle complicated column structures and projection profiles inside the segmented blocks. While not explicitly noted, a kind of RSLA algorithm is used for detecting connected components. For the determination of the thresholds for the grouping, height and distance histograms are used. For the border-line detection the process is described as follows: "First, border lines are extended upward and downward until they reach an element or an edge of a page. Next, two horizontal lines are created a t the top and bottom edges. If they crosses elements, the border lines are shortened until

they crosses nothing. " [66]. Next, an additional unification of the found segmented block structure is applied by checking the presence of border-lines between neighboured blocks in a left to right and top to bottom manner. If there is no border-line in-between and the second (right or bottom) block is not already part of a unified block, the blocks are merged to build a (new) unified block. If there are blocks that contain multiple types of elements (figures, text, lines...), the block is again segmented by a projection profile method which is not based on pixel density but on element type and therefore shows the presence of specific elements.

Akindele and Beläid [1] proposed an algorithm which collects the horizontal and vertical white gaps of a document combined with a threshold for excluding white space between characters, words and lines. The main advantage of the presented algorithm is its ability of extracting non rectangular "simple isothetic polygonal" blocks. The authors built an horizontal/vertical intersection table and walked trough this table by using a direction matrix.

Hao, Wang and Ng [62] described a nested segmentation approach where the input rectangles are spatially described by an adjacent block graph. The edge type is either diagonal, horizontal or vertical with an additional distance weight and with the nodes representing the boxes. The weights are then used for the cutting decision (more weight means more distance) for generating a so called L-S Tree (layout structure tree) which seems to be very similar to a X-Y Tree structure.

Saitoh, Tachikawa and Yamaai [123] developed an interesting segmentation and text reading order system that handles skewed documents and columns with non-rectangular shape. They used 8x8 pixel blocks as the base for the segmentation process. The string line direction is detected by various block distance calculations and projection profiles and the segmentation itself is based on a block classification for forming larger areas (classes: two text classes, horizontal/vertical lines, diagram, table, frame). The resulting structures are described as a hierarchical tree graph based on a node influence range which is initially set to the node-width and expanded according to spatially related nodes.

Antonacopoulos and Ritchings [6] described a white space algorithm which is able to detect also non-rectangular segments. After the skew detection with horizontal projection profiles, a dynamic vertical smearing value is calculated based on the distance between the upper baseline and the top small font line between two lines of text. The next step is the covering of the white space with rectangles that fit the longest horizontal directed white area and are merged, if the end points are close together and a predefined expansion threshold is not exceeded. The segmentation is done by building a graph with the white tile edges (not the white tiles themselves) and detecting graph cycles in an efficient sequential way for determining the contours of the non-white printed area.

Ha, Haralick and Phillips [60] presented an recursive XY cut algorithm which is based on bounding boxes of connected components instead of black pixels. The decomposition is recursively done using horizontal and vertical projection profiles at box level at each step that cuts at the largest profile valleys in both directions.

Normand and Viard-Gaudin [108] described a segmentation algorithm which is a two dimensional RSLA approach by using regular octagons as base elements. This elements are used for filling the white background and are the base for a hierarchical tree with adaptive tresholds used for the two dimensional smearing.

Wang and Yagasaki [141] proposed a multi step segmentation process using a tree structure that reflects the hierarchical inclusive relationship between components. The algorithm successively classifies the connected components and decides about merging or restructuring the tree. During this process, statistical parameters are collected dynamically and used for threshold values. At every stage, a kind of decision rules is applied to the classification.

Sivaramakrishnan et.al. [129] presented a zone classification algorithm using a decision tree with features based on horizontal, vertical and both diagonal directions. For each of the four directions line features based on runs of black foreground and white background pixels are calculated across

one single zone. These line features form the base for some of the zone features (also in each direction): number of foreground/background runs, background run length mean and variance, spatial mean and variance, fraction of black pixels, area and so on. This results in a feature vector with 67 fields used to define 9 different classes: two text classes, math, table, halftone, map/drawing, ruling, logo and other ([129]).

Sauvola and Pietikainen [125] showed an approach where the document is divided into small windows from which the features are calculated and the classification respectively the labeling is done. The following features are used: black/white-ratio, avg. black run length, vertical cross-correlations. The algorithm continues with an iteration of applying masks to the window map that propagates the dominant label.

Lovegrove and Brailsford [95] showed a document analysis based on PDF documents by comparing spatially related blocks. The authors concluded that over-segmentation with additional merging performs better than the opposite one.

Liu et. al. [93] used a quadtree with adaptive thresholds for the initial segmentation with the partitioning decision based on projection profiles. The spatial relationships are determined out of the quadtree handling non-uniform regions as well.

Kieninger [75] proposed a block segmentation approach for table extraction issues. It is based on word based clustering by merging vertically overlapping word blocks. Because of the errors which are made during this stage, an additional postprocessing process is executed. Wrongly isolated block and separated words are merged and merged columns are splitted.

Altamura, Esposito and Malerba [5] presented the WISDOM++ document analysis system based on a modified RSLA segmentation, a skew estimation component and a decision tree block classification (text block, horizontal line, vertical line, picture and graphics) together with the possibility of analysing multi page sequences and a user interaction for revising the classification.

Liang, Phillips and Haralick [89] used a probability based system that tries to find an optimal solution for a hierarchical partition described by a tree structure where the properties and semantic levels at each level are similar.

Mitchell and Yan [101] described an interesting approach called soft ordering that tries to apply a more logical reading order than simple left/right top/bottom approaches. The segmentation is done with a connected component analysis followed by additional pattern and context classification. The pattern blocking stage forms larger entities as input for the soft ordering algorithm which is based on a sigmoid function for considering also the height of the blocks.

Breuel [20] described a layout analysis based on a segmentation scale space. The primary idea is to look at all possible segmentations which are the power set of all connected components and to increment a minimal distance threshold resulting in the merge of two components, if their distance is below the threshold. The author refers to the correspondence assumption which implies that the logical layout hierarchy is paralleled in the document segmentation scale space ($page > column > paragraph > line > word$) and states that there is no reason for a general satisfaction of the assumption. An optimal segmentation respectively layout match is determined by a Bayesian framework.

Shi and Govindaraju [127] described a multi-scale approach with dynamic adaption of threshold values using multiple document resolutions for the computation of dynamic local connectivity maps (DLMC). The DLMC is generated by setting the background pixels to the run-length between two foreground pixels. The chosen method is the minimum of the horizontal and vertical distance . Therefore two maps are initially generated (in the size of the document) and combined afterwards.

Cao et. al. [29] presented a modified smearing algorithm based on a threshold that is dynamically adapted regarding the font-size. The preprocessing stage includes the component detection with the removal of non-textual components like tables and images. Table structures are detected using

the projection profile technique. For the image detection the authors filter the image and apply a binarization so the image becomes connected but text content remains separated.

2.1.2 Knowledge and Grammars

Higashino et.al [65] proposed a knowledge based system for document understanding using a form definition language which describes the layout as a set of rectangular coordinates.

Dengel and Barth [42] introduced a knowledge based document layout analysis system named ANASTASIL (Analysis System to Interpret Areas in Single-Sided Letters) with a geometric tree structure representing the possible layouts like a decision tree. The interpretation of a document is done by searching through the geometric tree until reaching a leaf node.

Fisher, Hands and D'Amato [54] demonstrated a rule-based segmentation system. Skew detection is done by applying a modified Hugh transformation to a reduced data set of the original image. The segmentation is based on a "horizontal - vertical AND horizontal" RSLA smearing sequence. The connected components (CC) are detected with a row- or run tracking method. By using features and specific rules the components are classified and segmented into text and non-text regions.

Esposito, Malerba and Semeraro [50] presented a document layout analysis system related to the RES knowledge based system with a new learning approach which integrates parametric and conceptual methods. Segmentation was done applying the RSLA algorithm with a rough classification of the segmented blocks (text, image, graphics, horizontal and vertical lines) using basic features like black-density and black/white frequency. After this step projection profiles are applied at block level for detecting text columns resulting in greater blocks which the authors named frames. The resulting document layout is transformed into a symbolic description including attributes and spatial relationships. This description forms the base for the automatic knowledge building, rule processing and classification of the documents using the RES system, "a problem-independent system for the automatic knowledge acquisition by training examples" [50]. The knowledge representation itself is done using an extended first order logic. Three experimental setups were conducted with a conceptual, statistical and an integrated method. The statistical method is based on a discriminant analysis reducing 93 picked features down to 6 using a stepwise variable selection. The 6 selected features were: maximum eccentricity of image blocks, standard deviation of the number of black pixels, standard deviation of the length of text blocks, minimum eccentricity of image blocks, symmetry along the vertical direction [50]

Lebourgeoise, Bublinski and Emptoz [[88] presented a rule based system using a modified horizontal RSLA algorithm applied to an image with a reduced resolution. The segmentation is implemented by four rules: the first and the fourth rule separate between text and images, the second rule separates connected but different blocks and the third rule extracts lines. For the paragraph merging, additional typographical and spatial rules are used.

Krishnamoorthy, Nagy, Seth and Viswanathan [84] described a grammar approach based on the X-Y tree algorithm and projection profiles. The main contribution of this paper is the combination of the segmentation task with the classification. The method starts with the segmentation using horizontal and vertical projection profiles in an alternating fashion. At every stage the profiles are converted intto a binary code using a threshold. The resulting threshold profile is immediately parsed with a context-free grammar in a four stage process: Atom Generation, Molecule Generation, Labeling, Merging. The first stage counts the length of the threshold profile strings and divides them into equivalence classes. The second stage is based on a LEX program and groups the atoms using a set of valid regular expressions. The third stage consists of a YACC parser which assigns labels to the molecules based on precedence and cardinality. The fourth stage simply merges continuous labels.

Bayer [15] showed a document analysis system based on the Fresco (Frame Representation of Structured Documents) Semantic Net. The system incorporates geometric, lexical and structural knowledge. Fresco is specialized in structure analysis and supports only the subclass and the has-part relation. The document knowledge is modelled by layout and logical concepts. Examples for layout concepts are text-blocks, lines or characters and spatial relations like lines-left-aligned or words-on-same-baseline. Logical concepts determine the application domain. The logical domain model in Fresco is dynamic and has a sub-class relation to the statically predefined layout model. The inference mechanism is a mixed top-down and bottom-up approach with the goal of generating "instances to layout and logic concepts until specific text portions of a document are interpreted " [15].

Conway [40] used a two dimensional grammar approach which is described as similar to a context-free string grammar. After the RSLA segmentation a so called chart parser is used until all segments are covered: "Chart parsers keep a record (called a chart) of well formed sub-strings which have been located and goals which are being investigated. This prevents the parser from repeating its efforts during a parse. Chart parsers also provide a flexible control structure which allows experimentation with a variety of search strategies" [40].

Kreich [83] presented the IDS (Intelligent Document Analysis) system based on a knowledge based approach about text and layout. There is no information given about the segmentation algorithm but the resulting hierarchical box structures and lines are the input for the knowledge base part called LyMona. The knowledge is splitted in domain and control knowledge. Examples for domain knowledge a "re definitions of shape and position of a business letter recipient and contents and syntax of a date. Examples of control knowledge are priorities of document parts and rules for the selection of hypotheses" [83]. Knowledge is presented by bundled semantic class structures with the possibility of multiple inheritance. The text classes use various lexica (names, companies), whereas the layout classes contain the geometric knowledge.

Rus and Summers [122] analysed the logical document structure by using a white background approach by generating a logical hierarchy which is based on the classification of base-text, tables, indented lists, polygonal drawings and graphs. The authors introduced an indention alphabet applied to the tree structure from the segmentation.

Niyogi and Srihari [107] developed a knowledge based approach for logical structure recognition named derivation of logical structure (DeLoS) system with a docstrum segmentation. The rule-based system consists of knowledge, control and strategy rules implemented in Prolog.

Esposito, Malerba and Semeraro [49] proposed a document layout analysis based on a knowledge based system named LEX (Layout EXpert) which was implemented in Prolog. The approach seems similar to their previous approach which is adapted from the RES knowledge based system (see [50]). The segmentation is done using a RSLA smearing algorithm. LEX is performing a global analysis for determining the larger areas like paragraphs, columns, etc. and a local analysis for the block grouping. The input parameters for each block are the top/left, bottom/right coordinates, block type (text, graphics,...), number of black pixels before and after RSLA application, number of horizontal white/black transitions ([49]).

Ishitani [69] [70] proposed a system for logical structure analysis which combines the layout analysis and the logical structure analysis and consists of five modules (typography analysis, object recognition, object segmentation, object grouping, object modification) with a strong interaction between the modules. The typography analysis does a classification into normal, indent, centred and previous to new textline. The object recognition does a classification into various logical objects (paragraph, title, formula, list) and is followed by the object segmentation for compound objects which are sent back again to the recognition stage. In the last two stages, objects are regrouped in the case of over-segmentation and modified in the case of errors followed by an additional processing.

Watanabe and Sobue [143] addressed the problem of complex layouts by using two views: an operator specification for the spatial relationships and a structure description for the physical features (position,length,...) which are represented by a layout knowledge tree. The authors used four kinds of operators: vertical and horizontal partitioning, hierarchical node (substructure) and terminal node (data field). The partitioning is executed by different kinds of dividers: blank, indention and line dividers.

Klink, Dengel and Kieninger [79] presented a hybrid approach by using textual and layout features with a common document structure recognition and a domain dependent logical labeling based on logical rules with fuzzy attribute matching (e.g. a block is also inside a region, if it slightly overlaps).

Malerba, Esposito and Altamura [96] described a first-order logic system which allows the user to correct the outputs and learns rules out of the applied user interaction based on horizontal/vertical splitting and grouping.

Kanungo and Mao [74] proposed a style directed document analysis system by using a stochastic language model for the physical layout and logical structures. The segmentation is executed by a probabilistic finite state automat.

2.1.3 Image Processing and Vectors

Bloomberg [17], [18] described an image processing approach based on filter windows (structuring elements) which are used for separating italic text or half-tone images.

O'Gorman presented an interesting method in [109] based on a vector approach. The algorithm starts with building five k-nearest vector pairs out from connected components. Each pair is described by a distance/angle tupel based on the component centroids which is one of the reasons why the algorithm is inherently independent from skew. So the document spectrum (docstrum) is a plot of all distance/angle tupel. Several histograms from the generated document spectrum are used for determining the orientation within line spacing and between line spacing.

Ittner and Baird [9] described a document analysis system with skew detection based on Fourier spectrum analysis, a greedy white-covers segmentation and language independent line orientation based on a minimum spanning tree using element centroids.

Iwane, Yamaoka and Iwaki [71] proposed a pattern classification approach by using low level image processing features. The input elements for the classification algorithm are connected black pixel components which are described by 9 feature values. Line height and line spacings are determined by projection profiles. The classifier is based on a feature vector dictionary which is described to be equivalent to a conventional but not hierarchical organized knowledge base and uses a minimum Euclidean distance scheme with reject thresholds.

Etemad et.al [51] used fuzzy and neural network techniques for layout independent page segmentation which is described as "not effected by skew, layout structure, text line orientations, font size or language" [51]. Blocks are built by scanning with a window W which is moved stepwise by w over the whole page with $W = 2w$. The basic idea is to build a vote matrix for each block and use weighted combinations from the neighbouring blocks. The algorithm works as follows: a multi layer feed forward neural net is used for building a soft block classification based on the interval [0,1] for every single class (text, picture and graphic). Feature vectors are obtained by using subbands of a wavelet transformation. The single class vote matrices for each block of width w are joined to a combined vote matrix. After a complete scan of the document the contributions of the neighbouring blocks are calculated using a "Vote Propagation Matrix" which defines how much a specific block affects its neighbour blocks. The analysis is done at different window dimensions starting with a low resolution. Higher resolutions are chosen, if the confidence level is not good

enough. The calculation of the multi-resolution analysis is again done with a matrix named "Cross Scales Vote Propagation Matrix".

Kise, Yanagida and Takamatsu [77] presented a white background segmentation technique which is based on background thinning (voronoi diagrams) for obtaining line chains. Unnecessary chains are removed by a minimum black pixel distance threshold and the difference of average line widths. An improved version is described by Kise, Sato and Iwata [78] by using additional conditions for the deletion of unnecessary chains.

Etemad, Doermann and Chellappa [52] used soft decision techniques based on multiscale feature vectors generated with a neural network. Segmentation is done using wavelet packets.

Cheng, Bouman and Allebach [37] described a multiscale approach based on a Bayesian network where the document is analysed with different resolutions using wavelet decomposition.

Chen, Jaeger, Zhu and Doermann [36] presented the DOCLIB software library for document processing which consists of various modules for the image processing and document analysis. It is extendable and contains the docstrum segmentation algorithm.

2.2 Web Page Analysis

2.2.1 HTML Tags and Wrappers

With the birth of the world wide web at CERN Conseil Europen pour la Recherche Nuclaire by Tim Berners-Lee ([16]) there was much research about hypertext and how to extract information out of web pages. The beginning of web page analysis was strongly related to query languages applied at the HTML source code. Konopnicki and Shmueli [81] described the SQL like W3QS language partly realized with standard UNIX programs. Lakshmanan, Sadri and Subramanian [86] presented the Weblog query language inspired by SchemaLog and Mendelzon and Mihaila [99] proposed another SQL like language named WebSQL implemented in Java.

Hammer et.al. [61] presented a tag based extraction implemented on top of the Python find command with a configuration file generating output in object exchange format which was orginally developed for the "The Stanford-IBM Manager of Multiple Information Sources" (Tsimmis) [35].

One of the earliest web page wrapper systems is proposed by Kushmerick, Weld and Doorenbos [85]. The Wrapper Induction Enviroment (WIEN) is based on start and end delimiters which identify the target data. For reducing the amount of manual interaction the authors introduced the wrapper induction method with the learnable HRLT (head-left-right-tail) wrapper class. The described algorithm takes a set of labeled web pages and returns a "consistent" HLRT wrapper. Consistent is defined as the capability of the wrapper to generate the labels which initially constructed the wrapper in turn. WIEN cannot handle missing items or permutations of attributes. Doorenbos, Etzioni and Weld[46] described another early wrapper system named ShopBot a domain-independent autonomous comparison-shopping agent using a combination of heuristic search, pattern matching, and inductive learning techniques. The learning phase for the generation of the domain model is performed off-line. The learning is based on identifying the search form, determining how to fill the form and interpret the search result page. For this task some test searchies are executed and the results form the input for the learning algorithm.

Brin [25] described an approach to extract author, title relations from the web by using Dual Iterative Pattern Relation Expansion. The used pattern is based on a simple regular expression. The expansion algorithm needs very few examples which are automatically expanded to much more examples (the paper described a sample set of only five books which was expanded to a list of over 15.000 books with minimal human intervention)

Lim and Ng [90] proposed a semistructured graph (SDG) which is generated by a high-level stack machine based on a push down automata. The HTML tags are classified into start/end tags, tags with optional or no end tag and so called unproductive tags.

Hsu and Dung [67] proposed the SoftMealy System based on a finite state transducer (FST) implemented in Java. Delimiters are replaced by the description of invisible separators using context tokens. The web page is tokenized and the separators are recognized by the FST by contextual rules. The wrapper induction is done by the generalization of this rules out from training examples. SoftMealy can handle missing or multiple attributes and variant attribute permutations and is a more general wrapper than the WIEN system by Kushmerick.

Muslea, Minton and Knoblock [103] introduced their STALKER wrapper induction system based on a sequential covering algorithm for hierarchical data extraction. A document or web page is described as a sequence of tokens which form the base of the so called embedded catalog (EC) formalism represented by a tree structure. The rules itself are based on a simple landmark grammar (SLG) and are generated by the STALKER induction algorithm (landmarks are, for example, HTML tags). The algorithm starts with a set of training examples and tries to find an optimal rule set. Optimal in this sense means a maximum covering of the positive examples. The remaining examples are processed again and so forth. STALKER needs fewer training examples than other previous algorithms.

Soderland [130] proposed the WHISK system using regular expression patterns for information extraction resulting in a wide range of possible documents including free text. WHISK is a supervised learning algorithm and needs some tagged training documents.

Again Lim and Ng [91] presented a heuristic based conversion from HTML to XML by classifying the HTML tags, like H1, H2 and layout related tags, according to their hierarchical meaning. Afterwards the former tag types are analysed through a precedence relation ship ($H1 > H2 > ... > (P, UL, ...) > ... > (TH, TD)$). The system was implemented in Java.

Baumgartner, Flesca and Gottlob [13] presented their LIXTO system capable of supervised wrapper generation and automated web information extraction based on the newly developed logical datalog-like language ELOG. Whereas forming the base of the system the normal user does not interact with ELOG directly but by using a sophisticate interactive user interface for the wrapper generation. The LIXTO system consist of the interactive pattern builder, the ELOG based extractor and the XML generator for the mapping. The interactive wrapper generation is based on hierarchical patterns which represents the default XML element names. When selecting a element on a web page similar instances are selected automatically based on a generalized DOM path. Additional filters and conditions could be applied on the patterns based on the tree structure and by using regular expressions which provides a great flexibility for the extraction process.

Crescenzi, Mecca and Merialdo [41] described their ROADRUNNER system based on a novel approach by using two HTML pages at a time to distinguish meaningful patterns from meaningless ones. HTML pages are regarded as a result of an automatic scripted generation with an underlying database. The problem is formulated as follows: given a set of sample HTML pages belong ing to the same class, find the nested type of the source dataset and extract the source dataset from which the pages have been generated. [41] . Therefore the extraction process is formulated as a decoding process. The matching is executed by the ACME technique for Align, Collapse under Mismatch, and Extract where HTML pages from the same class are compared by detecting string and tag mismatches.

Pan et.al. [110] devloped the DENODO platform, a semiautomatic wrapper with the DEXTL grammar utilizing various heuristics which result in a simpler language as traditional approaches. Access to web sources is applied by the navigation sequence specification language NESQL offering macros at browser level for easy navigation and the wrapper generation by ”means of examples”.

Liu, Grossman and Zhai [92] proposed an effective algorithm for mining data records in web pages called MDR. The algorithm is based on the observation that similar records are located in a particular data region, built by the same HTML tags and therefore reflected inside the DOM tree. The comparison is based on the concept of generalized nodes and a string edit distance with a threshold

Chang, Hsu and Lui [33] developed the IEPAD (Information Extraction based on Pattern Discovery) system one of the first systems which does not rely on user labeled examples. The systems consists of three components: pattern discoverer, rule generator and the extractor. The pattern discoverer uses a tree data structure named PAT tree which is a binary suffix tree for finding the maximum repeats representing the data records. Because there is probably more than one solution for the maximum repeats the results are presented to the user who selects one of the appropriate patterns.

Etzioni et.al. [53] developed their KnowItAll system for extracting facts (e.g. name of politicians) form the web. The input for the system is a small set of domain-dependent classes and facts from which the system starts learning by a bootstrap concept. The system needs no hand-labeled examples because of the domain-independent base and the bootstrap concept of the KnowItAll approach.

Banko et.al. [12] introduced a new extraction paradigm called Open IE capable of a fully single pass extration without any human interaction. They implemented the TextRunner system based on this paradigm and made a comparison with the KnowItAll system, a previous work by some of the authors. TextRunner consists of the self-supervised learner, the single-pass exractor and the redundancy-based assessor. The self-supervised learner does an automatic positive/negative labeling and applies a Naive-Bayes Classifier on the result. The single-pass extractor tags each word and generates relations with a lightweight noun phrase chunker. The redundancy-based assessor assigns a probability based on the number of distinct sentences.

2.2.2 Visual Web Page Analysis

Yang and Zhang [147] proposed a web page analysis approach based on the visual similarity of HTML objects. The visual attributes are parsed by a stack mechanism directly out from the HTML tags. For merging the smaller block elements into greater containers various fuzzy comparing rules are used.

Cai,Yu,Wen and Ma ([27],[28]) proposed their VIPS system, one of the first visual related web page analysis systems. The original target of the approach were small handheld devices but as later work shows the principal concept is much more general. The algorithm works in two stages: First the DOM tree is parsed and block elements are extracted according to the desired degree of coherence and various HTML tag characteristics. Second the separators are constructed by projecting each found box into the initial separators which are as large as the web page itself. Three main box operations are executed and the separator is either modified, deleted or splitted. This process is repeated several times until all boxes are processed. Through the definition of specific degree of coherence values different granularities could be achieved.

Burget [26] presented an interesting and simple approach by using a predefined presentation hierarchy as input. Regular expressions are used for the matching of the presentation hierarchy (respectively ontology) with the input HTML document. The matching is also based on visual characteristics like font size, bold/underlined, different colour and the heading level. The resulting logical tree is recursively corrected by using the visual weighted hierarchy.

Zhang, He and Chang ([148]) hypothesized the existence of a hidden syntax for database query web forms and interpreted the interface in a pure visual way using a visual language called 2P grammar. This grammar is capable for encoding both productions and preferences for the ambiguity resolution based on a kind of best-effort soft-parser.

Zhao et.al. ([149]) presented their automatic wrapper generation system for search engines. The system is based on both visual and tag features, introducing so called content lines which are described as horizontal text lines forming a horizontal line inside a section. Additionally the block shape respectively the indention lines are examined. After the feature extraction the record detection is applied by constructing a minimal SSR (search result records) subtree.

Simon and Lausen ([128]) introduced their ViPER system for detecting repeated record structures based on a modified genome alignment algorithm working on the HTML tag structure. The system is searching for maximum repeats in the context of data record alignment where the HTML document is regarded as a labelled unordered tree (there had to be at least two records to be present). The origin of the algorithm is the multiple (protein) sequence (genome) alignment from the bioinformatics area. Additionally a visual data region weighting is applied based on the visual location relative to the page center and the amount of coverage. This work is a good example of interdisciplinary research.

An visual interesting approach based on the VIPS system is proposed by Liu [94] where similar characteristics of data records or search results are collected and surrounded by a minimal rectangle. Based on the overlapping situation and the top positions of these rectangles a visual hierarchy is extracted. But because of the overlapping concept the algorithm needs at least two records.

Zhu et.al. [151] criticised the separation of the data record extraction phase from the attribute labeling phase and recommended a hierarchical conditional random fields method for the integration of the two areas. The authors stated that the separation of the two phases has some serious disadvantages like the error propagation from the record detection to the attribute labeling, the lack of semantics in the record detection, the lack of mutual interaction in the attribute labeling and the first-order Markov assumption. The base for creating the visual tree ist the VIPS system proposed earlier by two of the authors (see above [27],[28]).

Zhu et.al. [152] proposed the integration of both structure and text content understanding in a joint discriminative probabilistic model using Hierarchical Conditional Random Fields and Semi-Markov Conditional Random Fields.

Zhao, Meng and Yu [150] extracted search engine results using an analysis of the dynamic chaning part inside the web page. The authors described three problems which their approach solves: (1) Non-uniform section format problem, (2) Section-record granularity problem, and (3) Hidden section extraction problem. The method is based on tag and visual features with relations to a previous work (see above [149]).

Xue et.al [146] developed a web page title extraction system with a supervised machine learning approach comparing a DOM tree, a visual approach and a combination of both (the work is based on an earlier paper from Hu et.al. [68] where the visual aspect was not addressed). The experimental results indicate a similar performance between DOM tree and visual approach with an enhancement when using the combination of both. As a comparable baseline four different methods were used in the experiments: extract always the largest font, extract the first unit (a unit corresponds to a text line in the HTML document), evaluating the title tag, and the data from [68].

Baluja [11] demonstrated a segmentation based on Entropy reduction for the browsing on small screens. The principle idea is to use the DOM tree with the node positions itself for a decision tree representation and the learning algorithm.

3 Visual Algorithms and Data Structures

This chapter describes some of the important algorithms, data structures and methods developed over the time and tries to give a more detailed insight into the used principles. Selected papers reviewed before are probably mentioned again together with more detailed examples and/or figures for the specific algorithm as well as new literature presenting slight modifications or additions to the various methods. Of course this chapter is far from complete (if this is even possible) but we tried to mention the "famous" approaches together with some which are relevant to web page analysis in general and to our REDEVILA system. Many of the actual algorithms used for web page analysis are based on the older general document layout analysis methods. Graphic figures and sketches were either redrawn while trying to preserve the basic idea or newly designed for explaining a specific algorithm. For illustrative purposes we will present also some of the original figures from the cited papers.

Many approaches were originally developed for scanned documents and often operate on pixel level. But in fact there is of course the possiblity to apply (probably modified) versions of the algorithms also at web pages with their greater abstraction level regarding the basic elements like chars and full text blocks. One advantage of web page analysis in general is that there are (normally) no alignment problems which allows the direct application of projection profiles, X-Y Cut methods or other skew sensitive algorithms.

3.1 Segmentation

3.1.1 Image Processing

RSLA (Run-Length Smoothing Algorithm)

The RSLA (Run-Length Smoothing Algorithm) is sometimes also referred as "RSLA Smearing" (see [54] or [43]). The black foreground pixels are smeared into one direction similar to a bad rubber used on a drawing. The principal one dimensional algorithm applied to a binary sequence (e.g. black/white pixels) is shown by algorithm 3.1.

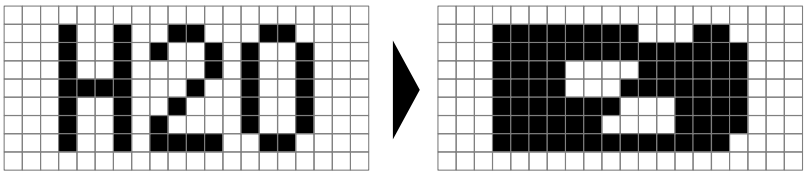


Figure 3.1: Run length smoothing example

The RSLA is therefore joining black pixels if the distance is below or equal a threshold t. If we have a sequence 000100000101000010000000011000 a threshold of 4 will result in the binary sequence 111100000111111100000001111 [145]. The final algorithm for segmentation purposes is applied

Algorithm 3.1 Simple one dimensional RSLA

Input: B: binary sequence; t threshold
Return: S:smoothed binary sequence

```
1:  function SIMPLERSLA(B, t)
2:      S ← []
3:      m ← 0
4:      k ← 1
5:      for k ← 0; k < |B| do
6:          if B_k = 0 ∧ m ⩽ t then
7:              m ← m + 1
8:          else
9:              if B_k = 1 ∧ m > 0 then
10:                 S ← S + [1]^m
11:             EndIf
12:             S_k ← B_k
13:             m ← 0
14:         EndIf
15:     end for
16:     return S
17: end function
```

line-by-line either horizontal or vertical. Figure 3.1 shows the horizontal RSLA effect on some chars and a digit with a threshold of 3.

There exist different variations of this algorithm regarding the orientation and the sequence. Wong, Casey and Wahl [145] presented a version in which they combine a horizontal and vertical RSLA through a logical AND (*horizontal* → *vertical* → AND sequence) forming a final segmentation (see figure 3.2). Fisher, Hands and D'Amato [54] used a *horizontal* → *vertical* → AND → *horizontal* sequence. Lebourgeoise, Bublinski and Emptoz [88] used only one horizontal smoothing.

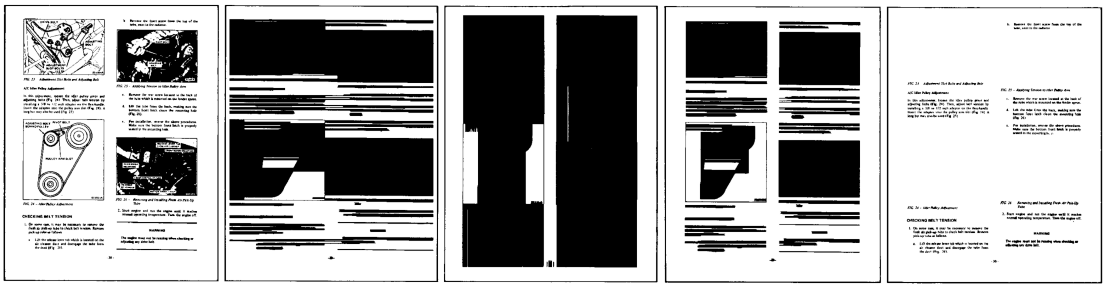


Figure 3.2: (from left to right): "(a) Block segmentation example of a mixed text/image document, which here is the original digitized document. (b) and (c) Results of applying the RLSA in the horizontal and vertical directions. (d) Final result of block segmentation. (e) Results for blocks considered to be text data (class 1)" (from [145], Fig. 2)

Another variant is described by Normand and Viard-Gaudin [108] that extends the RSLA approach by using regular octagons as base elements and makes it possible to operate on multi oriented structures. We will describe this 2D RSLA algorithm for simple squares based on the example given by the authors themselves: The background pixels are processed by replacing them by an index which expresses the largest possible square fitting the white background. The algorithm works in three steps: (1) the construction of the squares, (2) reducing the redundancy and (3) calculating the maximum squares (see figure 3.3). Afterwards a tree is constructed out of the background elements and adaptive thresholds are calculated for the final 2D smearing. In a strict sense, the initial first

part of this method is a kind of white background covering and could therefore also be used for background detection.

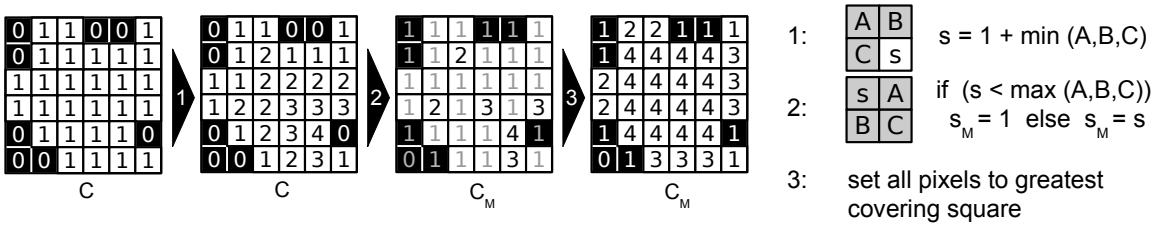


Figure 3.3: 2D RSLA with squares (after [108], Fig. 3)

Gatos and Papamarkos [56] proposed a fast segmentation by first decomposing the (black) image pixels into larger rectangular blocks which form the base elements for the RSLA. Shi and Govindaraju [127] presented an interesting variant based on a dynamic local connectivity map (DLCM). The authors look at the run-lengths of the white background determined by a horizontal and vertical map that are combined building the DLCM (see Fig. XX for the DLCM generation process)

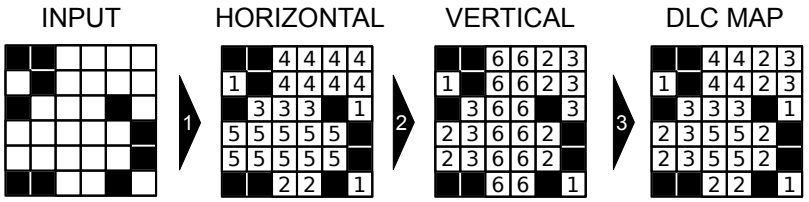


Figure 3.4: Generation of the dynamic local connectivity map [127]

Projection Profiles

Projection profiles are a simple method for gathering different characteristics of a document by generating a histogram. They are used for skew and char detection, for finding the line orientation (e.g. in Asian languages) and mainly for white separator detection during the segmentation stage (see also the RXYC algorithm).

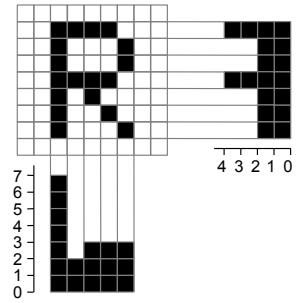


Figure 3.5: Basic principle of projection profiles

Figure 3.5 shows the basic principle of the projection profile approach whereas a horizontal projection profile $h(x)$ is defined as the sum of the black pixels projected onto the vertical x-axis of a binary

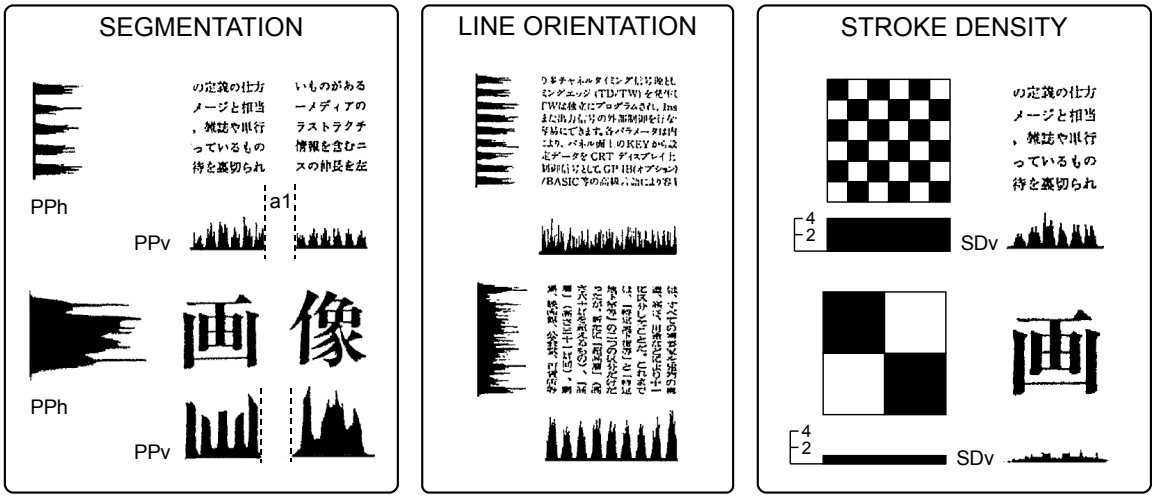


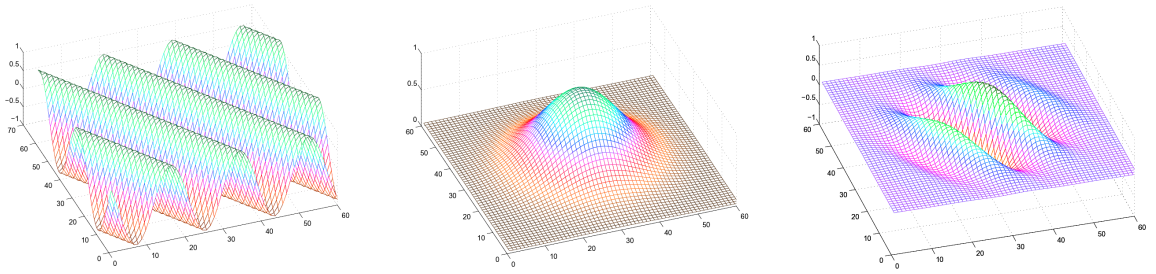
Figure 3.6: Different applications based on projection profiles (after [2])

image $f(x, y)$ and the vertical projection profile $v(x)$ defined as the sum of the black pixels projected onto the horizontal y-axis ([87]).

Different applications for projections profiles where presented by Akiyama and Masuda [2] themselves. One important problem in asiatic languages is the detection of line orientation shown in the middle of figure 3.6. The stroke density feature at the right of figure 3.6 is used for distinguishing large isolated chars from text blocks by counting the number of black/white inversions either horizontal or vertical.

Gabor Filters

Gabor filters in general are widely used in the image recognition domain. In contrast to simple Fourier fransformation they provide support for sepearating spatial related features.



(a) (b) (c)

Figure 3.7: "Gabor filter composition: (a) 2D sinusoid oriented at 30° with the x-axis, (b) a Gaussian kernel, (c) the corresponding Gabor filter. Notice how the sinusoid becomes spatially localized." (from [114])

A Gabor filter is obtained by the multiplication of a sigmoid function with a Gaussian function and could be defined for the two dimensional case by

$$g(x, y, \theta, \phi) = exp(-\frac{x^2 + y^2}{\sigma^2})\ exp(2\pi\theta i(x\cos\phi + y\sin\phi)) \tag{3.1}$$

where θ is the spatial frequency and ϕ the orientation. Figure 3.7 shows the basic principle for the two dimensional case (see also [114]).

Qiao, Li, Lu and Sun [117] described a text extraction algorithm based on Gabor filters. Figure 3.8 shows an example from the paper starting with the original document image on the left, some of the Gabor filters, the filtering result and the final text extraction result on the right.

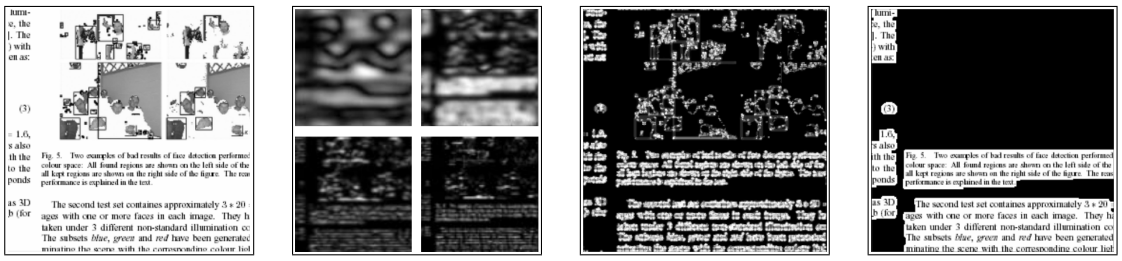


Figure 3.8: Text extraction with Gabor filters (from [117], Fig. 2, 3, 4)

3.1.2 Trees

X-Y Tree and RXYC (Recursive X-Y Cuts)

The X-Y Tree (or XY Tree) is a tree data structure described by Nagy and Seth [105], where each node corresponds to a rectangular box and each successor of a node is obtained by strictly alternating horizontal X-cuts and vertical Y-cuts with the leaves presenting the basic elements of the document. The X-Y tree is a further generalization of the k-d tree data structure similar to the treemap or the puzzletree ([124]). Figure 3.9 shows the basic principle of the X-Y Tree generation process.

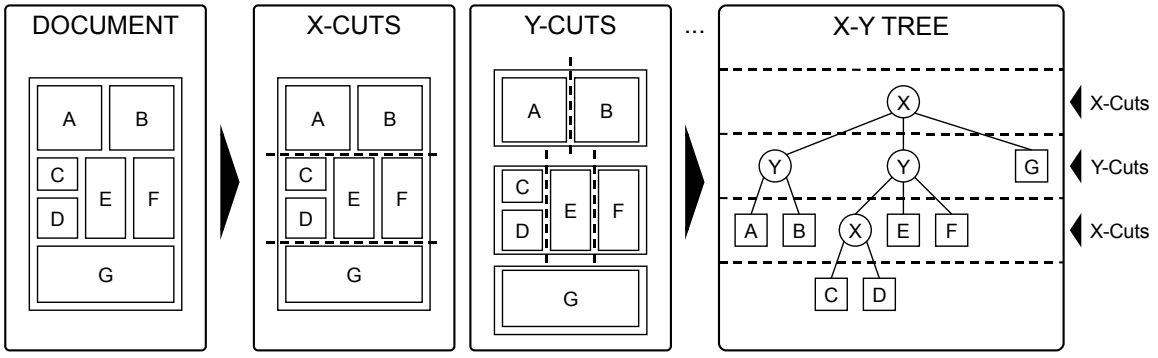


Figure 3.9: X-Y Tree generation

There exist different versions of this data structure, for example, Cesarini et.al ([32], [31]) described a modified XY Tree (M-X-Y) that handles not only white space but also lines (e.g. for tables). In the strict sense, this is not a modification of the X-Y Tree data structure itself, but a modification of the cut strategy. Marinai et.al [98] used the X-Y Tree for document retrieval based on tree edit distance together with a grammar for reducing the tree. A very similar data structure is presented by Hao, Wang and Ng [62] which they called layout structure tree (L-S Tree). It is described as an ordered labeled tree that consists of basic non-composit and horizontal/vertical composite nodes. The successors of the horizontal nodes are ordered from top to bottom and for the vertical nodes there is a left to right ordering.

While presented together with the X-Y Tree by Nagy and Seth [105] and also in [106], the recursive X-Y Cut method could be seen as a separate algorithm for finding cut positions based on projection profiles. The X-Y Tree is the data structure for holding the resulting horizontal/vertical rectangular structures. The term RXYC is used by Wang and Srihari ([140] explicitly for the projection profile based determination of the cuts. We will refer to the term RXYC in a more general way as an algorithm for applying (alternating) horizontal and vertical cuts in a recursive manner.

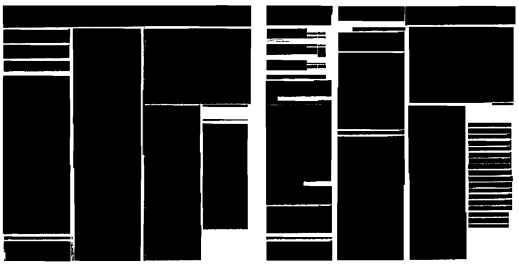


Figure 3.10: RXYC (left) compared to RSLA (right) – (from [140], Fig. 1 (e)(d))

Wang and Srihari ([140] presented a good example for a comparison between the RSLA and the RXYC method. Figure 3.10 shows a newspaper image with RXYC segmentation on the left and RSLA segmentation (with a bounding box calculation) on the right. We can see that RSLA produces a finer segmentation compared to RXYC.

Sylwester and Seth [136] proposed a modified version with an adaptive threshold which they called adaptive RXYC (ARXYC). The layout structure representation of Watanabe and Sobue [143] could be seen as a kind of X-Y Tree structure but targeted at layout analysis.

3.1.3 Vector approaches

Vector approaches are generally based on drawing vectors around connected components and analyzing the neighbourship relations. One advantage of this approach is the inherent skew robustness because the vector orientation follows the orientation of text lines. Also more complex text and non-text shapes could be expressed and analyzed.

Centroid Vectors (DOCSTRUM)

The Docstrum (document spectrum) approach from O'Gorman [109] is based on a k-nearest neighbour clustering. A vector is generated for each of the k-nearest neighbour pairs which describes the distance and the angle between the two component centroids. Generally a value of $k = 5$ works for most standard situations because it describes four neighbourship relations and one for redundancy. Depending on the needs the value could be decremented for simple text lines or incremented if there is a greater line-spacing. The docstrum is a plot of all distance/angle tuple with the angle quantized to $[0, 180)$ and additional mirroring because the vectors are undirected. As figure 3.11 shows, the resulting Docstrum plot is therefore symetric. Several histograms from the generated document spectrum are used for determining the orientation within line spacing and between line spacing. [109]

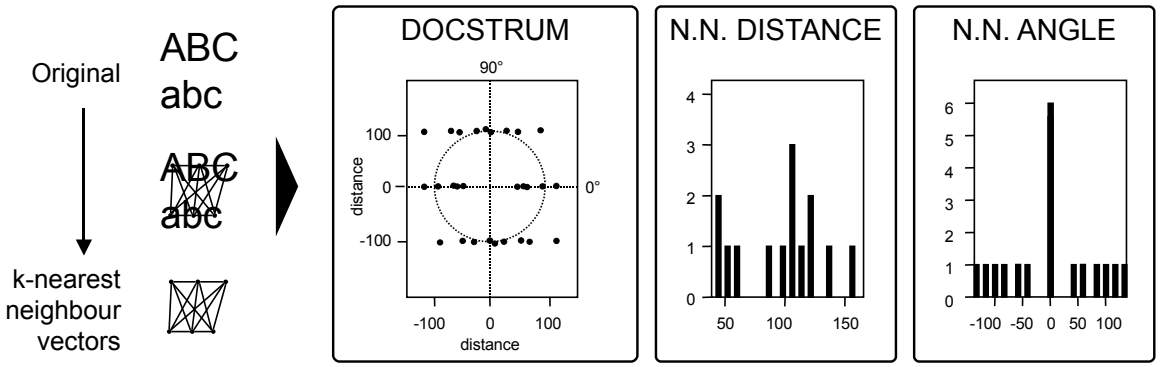


Figure 3.11: k-nearest neighbour vectors (left) and the docstrum with distance and angle histograms

Voronoi Diagrams

Kise, Iwata and Matsumoto [76] provided a short overview of the application of Voronoi diagrams to page segmentation including an example of a simple text (see also the top area of figure 3.12). Voronoi diagrams are basically created by a line which is normal to the vector defined by two points. Area Voronoi diagrams are created through a set of non-overlapping figures by using points from the figure contours and deleting all edges between points which belong to the same connected component. Somehow similar to the Delauney triangulation a neighbour graph is generated by connecting the center points of components which share an edge inside the Voronoi diagram. Different to the Docstrum approach with its definition of k there is no need for parametrization. The segmentation itself consists of finding the correct edges from the Voronoi diagram as shown in figure 3.12 and utilizes the distance information in the neighbour graph. [76]

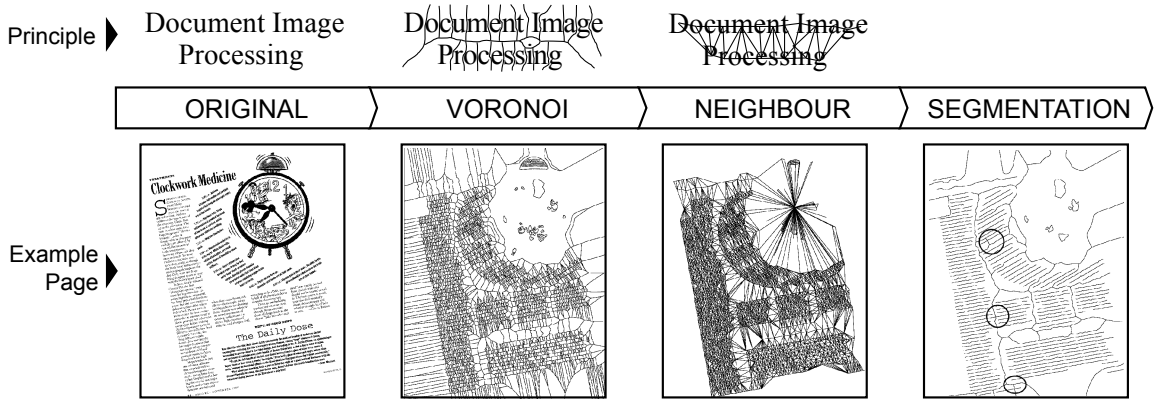


Figure 3.12: Area Voronoi diagram with neighbour graph and segmentation result (after [76])

An earlier application called background-thinning was presented by Kise, Yanagida and Takamatsu [77] where the white background is thinned with 4-connectivity based on a fast image processing algorithm. This approach has a more pixel oriented view at the document but is principally similar to the area Voronoi approach.

3.1.4 Blocks and Contours

We will refer to the term block analysis as for algorithmic methods, whose basic element is a block in the sense of a greater entity than pixels or single chars. Various spatial relations are examined and described based on this block level.

Whitespace Covering

Baird [10] presented an approach based on covering the white background with maximum rectangles. A rectangle is maximized if it touches either black components or the document edges on each side so it could not be expanded any more without containing one or more black foreground components. Figure 3.13(a) shows the basic principle of finding all maximal whitespace covering rectangles with two black points. The final rectangles are displayed again outside the bounding box. Baird described the further extension of the basic point oriented approach to a black forgreound rectangle based algorithm by separately considering the sides of the rectangles. After finding all the maximum rectangles a subset called cover-set is selected by using domain specific knowledge, e.g. favoring high aspect ratios because of the column structure of the target documents (see figure 3.13(b) for a final segmentation example).

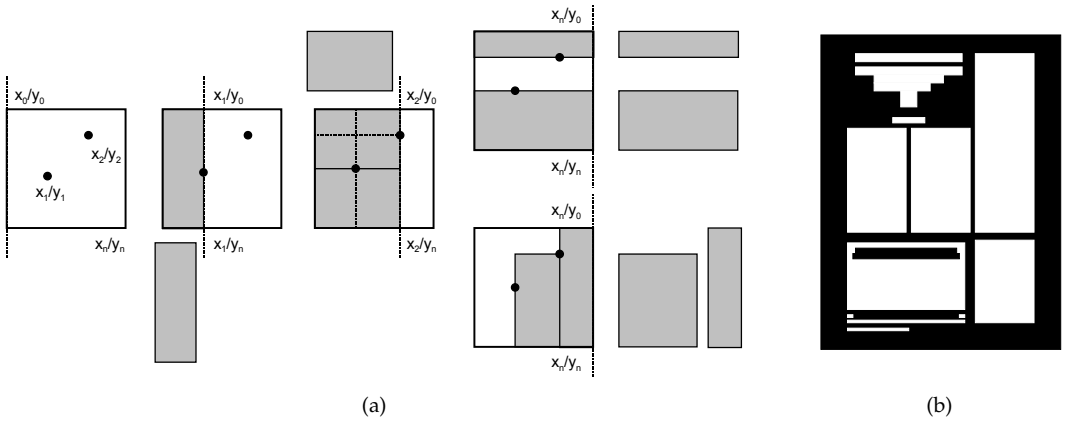


Figure 3.13: (a) Basic principle for finding all maximal white background covering rectangles, (b) A segmented Scientific American page which generates 11212 maximum rectangles with a cover set of 112 (after [10])

A variant of background covering is described by Waked [139] using diagonal white runs (or druns as he called it) for the generation of square boxes. The final segmentation is done by vertical scanlines between two overlapping squares and additional horizontal scanning. If a black pixel is found, the next two candidates are selected. Although perhaps a similar result could be obtained by simple vertical and horizontal scanning of the whole page, the idea seems to be unconventional.

White Space Tracing

White space tracing refers to approaches that try to collect the white background by using a kind of scanning procedure. An example for such an approach is described by Akindele and Belaid [1] based on contour tracing where an additional tracing algorithm is applied after a segmentation process. The authors built an horizontal/vertical intersection table where each element is either ”1” indicating a crossing of the segmentation rectangles or ”0” if no intersection appears. Furthermore

a walk through this table is applied by using a direction matrix and starting a search from each "1" entry. As mentioned before the first stage of the 2D RSLA from Normand and Viard-Gaudin [108] is also a background covering algorithm.

Maximum Empty Rectangles

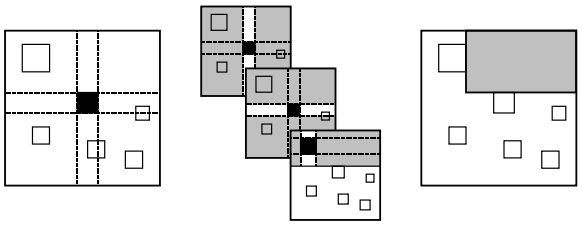


Figure 3.14: Finding maximum rectangles (after [24])

Breuel [21] described a whitespace covering algorithm based on maximum empty rectangles. Given a set of rectangles on a plane an evaluation function is defined which is maximized according to an ordering criteria, e.g. the area (see also figure 3.14). Breuel proposed an additional requirement by setting a minimum number of components which should be bounded at the major sides to make sure that the selected rectangles support the layout interpretation. The algorithm could be extended by the use of covering rectangles which are not parallel to the axis [22].

The advantage compared to the full white space covering is the easier implementation. Algorithm 3.2 shows the basic principle by selecting a rectangle (e.g. the most centered one) which gives four possible solutions for the maximum rectangle problem: left, right, above or below the initial rectangle. The additional covered rectangles are scored by a quality function and the associated rectangles are put into a queue. These steps are repeated until the queue is empty.

Algorithm 3.2 Finding the optimal whitespace rectangle (from [21])

Input: *bound*: outer bound
Return: *rectangles*: collection of rectangles

```
1:  function FINDWHITESPACE(bound, rectangles)
2:    queue.enqueue(quality(bound, rectangles))
3:    while not queue.isEmpty() do
4:      (q, r, obstacles) ← queue.dequeueMax()
5:      if obstacles = [] then
6:        return r
7:      EndIf
8:      pivot ← pick(obstacles)
9:      r_0 ← (pivot.x_1, r.y_0, r.x_1, r.y_1)
10:     r_1 ← (r.x_0, r.y_0, pivot.x_0, r.y_1)
11:     r_2 ← (r.x_0, pivot.y_1, r.x_1, r.y_1)
12:     r_3 ← (r.x_0, r.y_0, r.x_1, pivot.y_0)
13:     sub_rectangles ← [r_0, r_1, r_2, r_3]
14:     foreach sub_r ∈ sub_rectangles do
15:       sub_q ← quality(sub_r)
16:       sub_obstacles ← [list of u in obstacles if not overlaps(u, sub_r)]
17:       queue.enqueue(sub_q, sub_r, sub_obstacles)
18:     end for
19:   end while
20: end function
```

VIPS

Cai et.al. described a DOM tree based approach ([27], [28]) for the VIPS (Vision based Page Segmentation) system including a precalculated degree of coherence for the expected segmentation resolution. The difference to the other background covering algorithms is the use of maximum separators spanning the whole width or height of the parent area. This is probably somehow similar to the RXYC method described above because also the cutting lines are spanned across the whole width/height. Also the so called "L-Shapes" [100] could not be segmented by both methods. Figure 3.15 shows the concept with an example page from Yahoo.

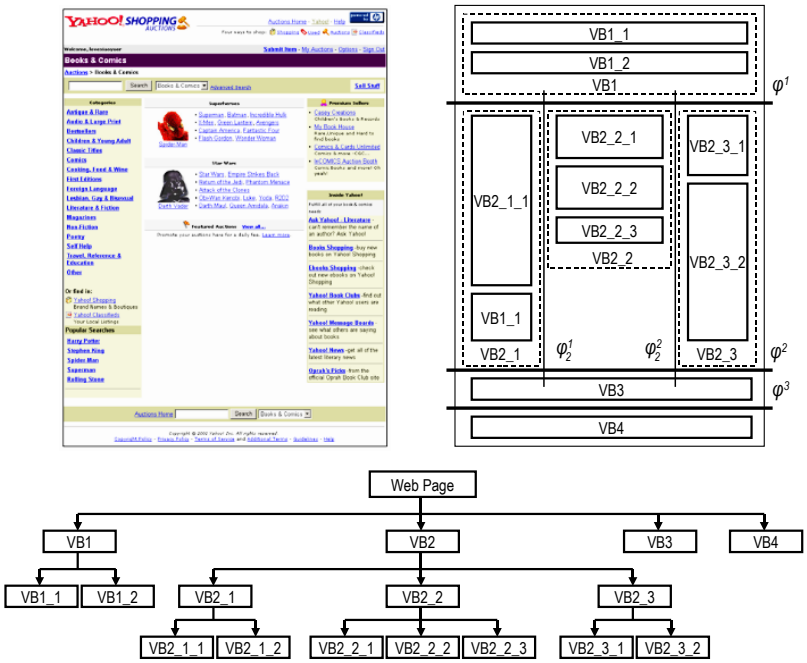


Figure 3.15: Layout and segmentation tree (after [28])

Block Clustering

Block clustering refers to black foreground algorithms which try to merge different block elements according to a neighbourship relation. Kieninger [75] described a clustering algorithm based on word blocks. After the clustering, additional merging and splitting operations are applied because of initial errors made by the pure clustering algorithm.

Segmentation as Entropy Reduction

Baluja [11] described an interesting segmentation approach by applying a decision tree approach on the DOM node positions. Therefore the segmentation process is formulated as an Entropy reduction problem. Figure 3.16 shows the basic principle with the original on the left, the segmentation in the middle and a random colored result.

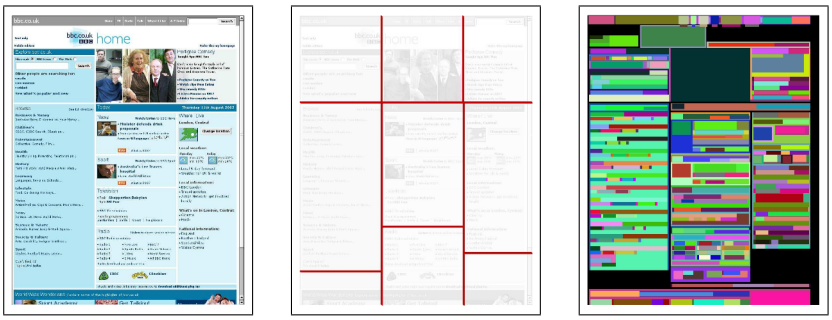
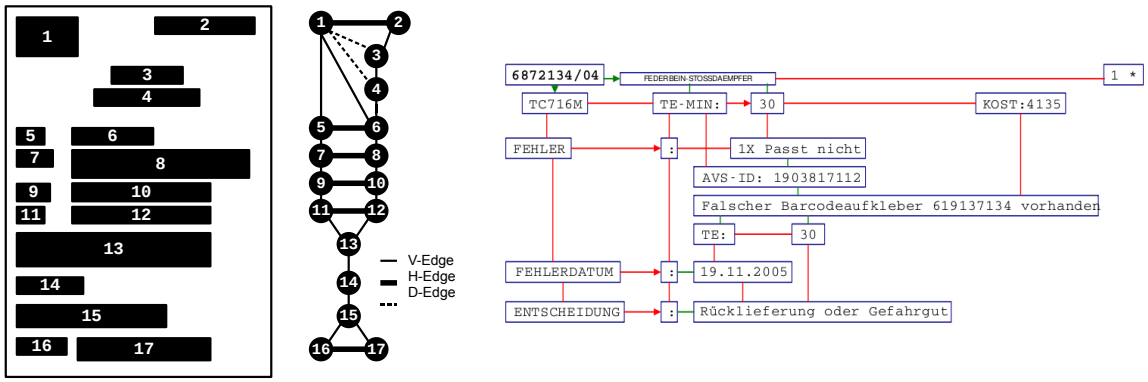


Figure 3.16: Segmentation as Entropy reduction (from [11])

3.2 Spatial Relations

Adjacency Graphs

Adjacency graphs are structures where the edges reflect spatial relationships between the nodes. Depending on the task, different spatial information is stored. Hao, Wang and Ng [62] showed an adjacent block graph which is a weighted undirected graph with the document blocks as nodes, the edges representing the relation (horizontal, vertical, diagonal) with an additional weight parameter reflecting the distance (figure 3.17(a)). Another variant is the Block Adjacency Graph described by Jain and Yu [72] where the pixels are grouped into larger blocks.



(a) (b)

Figure 3.17: (a) "Adjacent block graph from a memo" (after [62], Fig. 4), (b) "Sub-graph for one record (wrapping instance) [...] Note that edges with arrows represent superior-to-inferior relationships." (from [64], Fig. 2)

Hassan and Baumgartner [64] used a graph structure for describing the spatial relations inside PDF documents. The initial graph describes the adjacency relations and is filled successively with additional information like alignment and logical ordering (figure 3.17(b)).

Another graph approach is introduced by Kovacevic et.al. [82]. The nodes of the *Visual Adjacency Multigraph* represent simple basic html objects like text or images and the edges reflect the spatial relationships like left, right, above and below. Figure 3.18 shows the basic principle. Note that the VAM graph is splitted into four distinct graphs for presentational purposes. The graph is then used

for generating heuristics regarding horizontal or vertical link lists, titles and content text blocks. The classification is applied by a neural network approach.

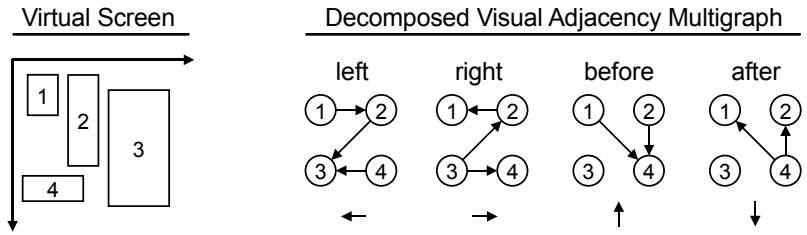


Figure 3.18: *Visual Adjacency Multigraph* with the virtual screen on the left and the decomposed graph on the right (after [82], Fig. 1)

Topological Grids

Topological Grids are a simple grid data structure optimized for spatial reasoning. An example is the double topological grid mentioned by Gatterbauer et.al. [59] which was developed for box based visual web table extraction. Figure 3.19 shows the basic principle with the minimal grid structure superimposed onto the web page and the double coordinate system from Top/Left and Bottom/Right.

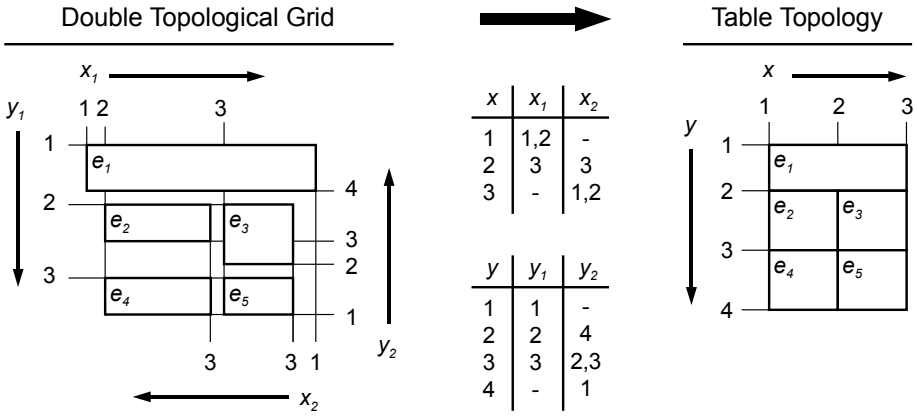


Figure 3.19: "The double topological grid allows to separate the step of locating a table and its composing logical elements from recognizing its topology." (after [59], Fig. 6)

An extension of the double topological grid is the multi topological grid developed for the REDEV-ILA system as a light weight alternative to the adjacency multi graph. It is based on the assumption that a precalculation of all spatial relationships is not necessary in general. Instead, the application of rules is executed serially on a block by block basis. Therefore the multi topological grid provides an easy infrastructure for block based spatial reasoning (for further details see chapter 5.6).

Logical Calculus

Qualitative spatial reasoning is an area of artificial intelligence with a wide area of applications (e.g. graphical information systems and navigation). The probably most fundamental concept of space

is topology but because of the qualitative centric view it has also some disadvantages. Cohn and Hazarika [38] gave an overview over the area of the qualitative spatial reasoning field which tries to represent not only common-sense knowledge but also the underlying abstractions with spatial semantics (see also [39]).

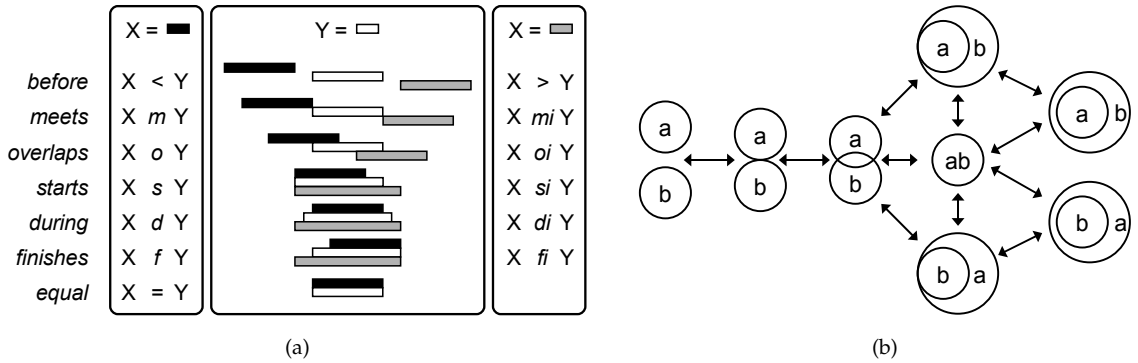


Figure 3.20: (a) Allens temporal intervals (after [4]) (b) RCC-8 relations and transitions (after [38])

Two important approaches are the temporal intervals of Allen [4] (see also figure 3.20(a)) and the region connection calculus from Randell, Cui and Cohn [120] with its variants RCC-8, using eight mutually exhaustive and pairwise disjoint relations (see also figure 3.20(b)), and RCC-5 without the boundary considerations from RCC-8.

2P Grammar

#	*Production Rules*	*Visual Patterns*
P1	**QI** ← HQI | Above(QI, HQI)	
P2	**HQI** ← | Left(HQI, CP)	
P3	**CP** ← TextVal | TextOp | EnumRB	
P4	**TextVal** ← Left(Attr, Val) | Above(Attr,Val) | Below(Attr, Val)	Attr Val
P5	**TextOp** ← Left(Attr, Val) and Below(Op,Val)	Attr Val
P6	**Op** ← RBList	Op
P7	**EnumRB** ← RBList	RBList
P8	**RBList** ← RBU | Left(RBList, RBU)	RBList RBU
P9	**RBU** ← Left(radiobutton, text)	radiobutton text
P10	**Attr** ← text	
P11	**Val** ← textbox	

Figure 3.21: "Productions of the 2P grammar" (after [148], Fig. 6)

The 2P grammar by Zhaang, He and Chang [148] is able to express simple spatial relationships for their hidden syntax approach regarding web forms. Figure 3.21 shows the basic visual patterns together with the production rules.

3.3 Reading Order

Soft Ordering

Mitchell and Yan [101] introduced a soft-ordering algorithm based on a sigmoid function (formula 3.2) by dominating left ordering depending on the block height where *avh* is the average pattern height and *minh* is the minimum height of two compared blocks. The approach is targeted at the printed document domain by considering heights and large column structures.

$$limit = \frac{avh}{1 + \frac{4}{avh} \mathrm{e}^{avh - minh}} \tag{3.2}$$

Topological Sorting

Breuel [24, 23] proposed an ordering based on topological sorting with the following two criteria for the previous partial ordering: *"(1) Line segment a comes before line segment b if their ranges of x-coordinates overlap and if line segment a is above line segment b, (2) Line segment a comes before line segment b if a is entirely to the left of b and if there does not exist a line segment c whose y-coordinates are between a and b and whose range of x-coordinates overlaps both a and b"* [23].

Figure 3.22 illustrates this concept. The diagonal ordering approach described in chapter 5 is somehow similar to this method but needs only one simple pairwise comparison, without the consideration of an additional block *c*.

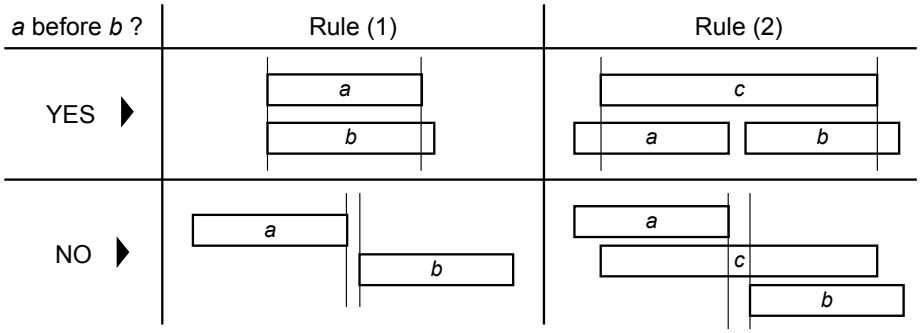


Figure 3.22: Topological ordering with partial ordering criteria (after [23])

XY Cut Ordering

Ishitani [70] presented an improved X-Y Cut algorithm which includes reading order analysis by using a pre grouping process before applying the cuts. Meunier [100] presented an improved deterministic version using an optimization approach with a score function.

The algorithm starts with the enumeration of all possible horizontal cuts inside a block. Afterwards all possible vertical cuts inside the resulting potential sub blocks are enumerated. A set of horizontal cuts is selected which satisfies the score function best and the cutting operations for both the horizontal and vertical case are executed. The score function favors vertical cuts spanning accross multiple blocks because of the printed document target with its common column structures. Also the inverse of the block distance is included so that the probability for merging nearer blocks is higher.

Figure 3.23 shows the basic principle with the initial enumerated cuts on the left and the final choices on the right. To reduce the computation complexity dynamic programming techniques are applied.

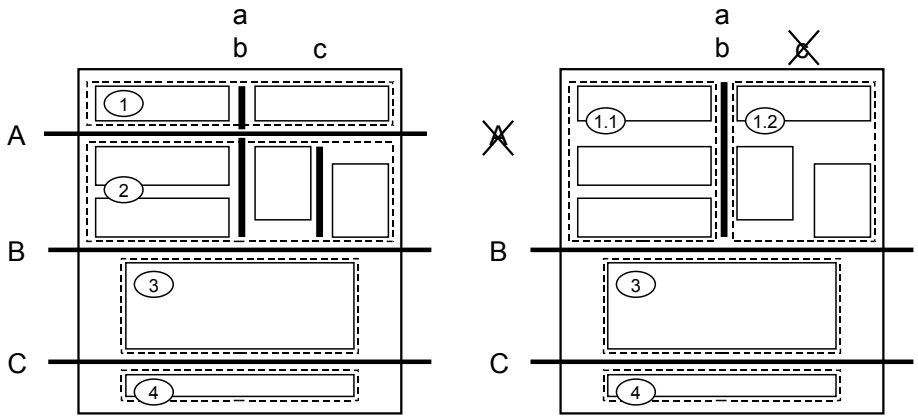


Figure 3.23: Optimized XY cut ordering (after [100], Fig. 3)

4 Building a Web Page Ground Truth

One difficulty in the area of web page extraction and especially for visual based methods is the lack of standard corpora for testing. One reason is the volatile and elusive nature of the web in general and the fast changes in web technology. Compared to print products and general document analysis there are also not that many kinds of standard algorithms which would allow the required degree of standardization of the datasets.

Although there exists a standardized dataset for the general document analysis (namely the University of Washington Document Database), another problem is introduced, because the experimental results could only be as good as the underlying document repository. This raises the danger of missing real world situations and optimizing tendencies which would make a comparison possible but hence not very meaningful. The other solution would be the generation of individual datasets with similar problems but the lack of comparison possibilities because of the private nature of these datasets which are often not published (or removed over the time).

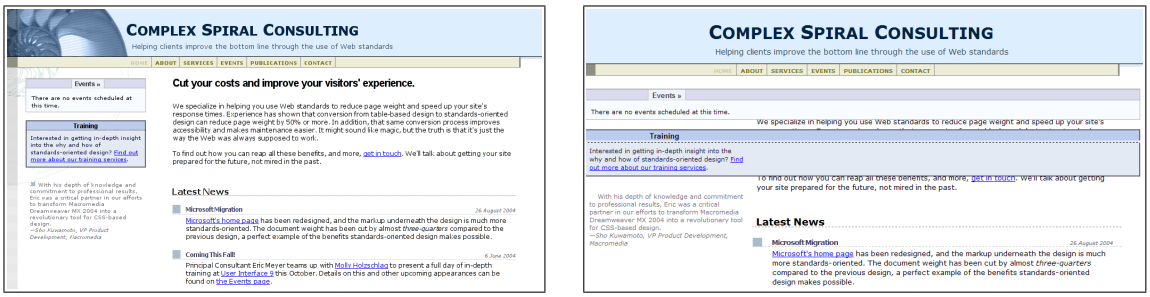
Due to the very different requirements of the various web page analysis approaches (general layout analysis, web table extraction, extraction of search results, web page titles, news etc.) and the fast evolving web page technologies, it would be difficult to define a standard dataset. While the lack in the web page analysis domain is bemoaned in the literature (e.g. [116]), such a standard dataset would probably be quickly outdated or not representative for a new scientific web page analysis approach which was not considered during the dataset generation. This is articulated for example by Qi and Davidson who stated: *"How can a truly representative dataset with these properties that is multiple orders of magnitudes smaller than the actual Web be selected?"* [116]

Due to these dynamic needs there seems no way beside the individual dataset generation. To overcome the problem of optimizing a specific dataset, it is absolutely necessary to make the generated datasets publicly accessible. This additional measure would give the key advantage over a main standard dataset. Researchers might be able to download other experimental datasets probably generated for similar problems making the scientific approaches better comparable. Maybe this could result in an evolvement of a kind of standardization for some web page analysis domains without debasing the flexibility.

What is needed is a scientific initiative for providing a public available disk space for individually generated test data sets. Unfortunately this raises questions regarding copyright issues which are not properly addressed to date. Maybe a restricted access for students and scientists would solve this particular problem. Also the web page images (notably corporate logos and humans) could be distorted maybe automatically through an image recognition algorithm.

4.1 Making Web Pages Permanent

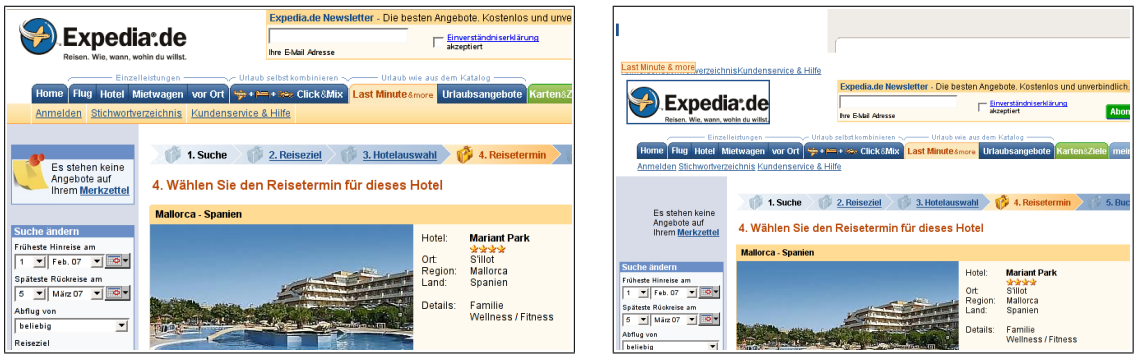
Of course the requirements of generating a web page dataset whether standardized or individual raises the question of the web page storing. Probably surprising, this is not an easy task as examined by Pollak and Gatterbauer [112]. In the early days of the world wide web simply the HTML code was saved (e.g. [142]) also because many algorithms were based directly on the textual HTML representation. With the upcoming visual based algorithms this is not the case any more. Hence it is necessary to provide a visual exact copy of a web page.



(a) (b)

Figure 4.1: (a) Original webpage, (b) Web page saved with Internet Explorer 6.0[1]

Unfortunately this is a difficult task because many web pages are not physically present as an entity but generated dynamically by a web server on request. Java applets and various browser bugs make it rather difficult to get an equal visual presentation between the online version and the local copy. Another problem is the use of dynamic JavaScript and various AJAX techniques. Due to on- demand user triggered visual interaction it is much more difficult even to define exactly what a single web page is, not to mention of making a local copy of the web page. As figure 4.1 and figure 4.2 shows, also the main browsers have serious difficulties in generating a local copy probably because this task seems not that important for many people. With more and more information available on the web this will maybe change in the future as the success of the Scrapbook Firefox extension shows (see also [113]).



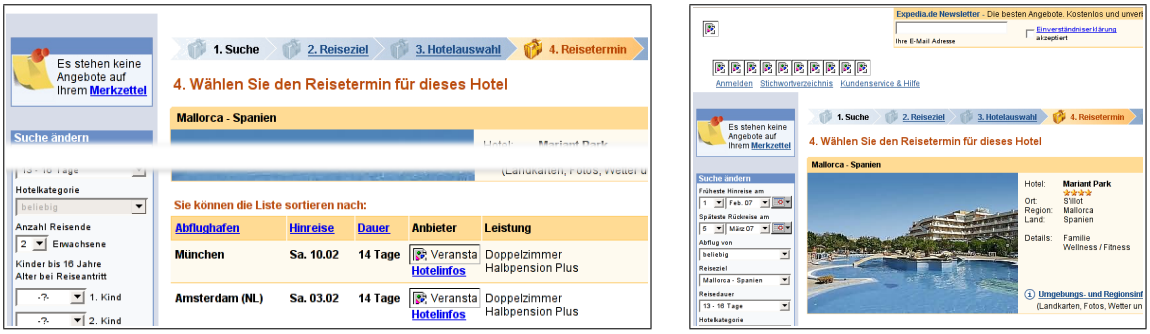
(a) (b)

Figure 4.2: (a) Original webpage, (b) Web page saved with Mozilla Firefox 2.0[2]

One possible solution would be the use of a proxy approach where the communicated data itself is stored. This data could then be accessed by different browsers and would theoretically produce the same result as provided by the original web server. But there is the possibility that the web server sends different content according to the detected client. While this is not very usual for standard web sites, it is, for example, used for small hand-held devices. Besides this, the proxy approach has some disadvantages regarding the need for a special proxy software and the distributed storage of web page parts, if different URLs or domains are used because a proxy could not separate the HTTP requests in such a manner. And as figure 4.3 shows there are problems with the dynamically

[1] http://complexspiral.com
[2] http://www.booking.expedia.de

created and internet connection dependent menus or travel agency logos not only with proxies but also with website downloaders.



(a) (b)

Figure 4.3: Webpage from figure 4.2 saved with (a) WWWOFFL personal proxy and (b) httrack website downloader

Originally HTML was never intended to provide a visual exact reproduction. Instead it was a structural description language for describing the logical hierarchy of a document (e.g. H1, H2 etc. tags). Over the years the needs changed, probably also due to the further development of hardware and graphical capabilities and HTML was extended to provide more and more visual layout capabilities. The uncontrolled growth of company specific HTML tags led to the introduction and standardization of CSS.

The term copy respectively reproduction comprises a philosophical component which normally represents no problem due to the determined digital representation. But in the case of HTML/CSS and different browser products and versions, different screen resolutions and often the absence of a clear well defined web page entity, this philosophical component arises. Especially when it comes to the generation of a ground truth or test data with positional information like in the case of visual web page analysis.

Figure 4.4: Example for resolution dependent results. Starting with a width of 1280 pixels (minus the width of the REDEVILA interface), the window width is successively reduced resulting in different classification results regarding the noisy segments (see the large "N")[3]

For example, the tag positions are often determined by the size of the browser window. When these positions are interpreted by another researcher, probably together with the corresponding web page, the original window size (together with the browser product and version) has to be communicated for an exact reproduction (see figure 4.4 for an example). Another problem are the installed fonts. Web pages normally give font alternatives or only the font family because no web developer knows in advance which fonts are installed at the target machine (e.g. verdana, sans serif). But if a researcher is opening a web page and has some fonts not installed, the positional information and visual appearance could not be exactly interpreted and reproduced.

4.2 Saving Problems

We will present the JavaScript double execution problem as an example for a saving problem (for additional saving problems and in-depth analysis refer to [112]). The following simple JavaScript code will display "Hello End":

But when reopening the web page after saving with Firefox (Web Page, complete) the text changes to: "Hello Hello End". This is caused through the double execution of the JavaScript code. During the first load the HTML code is inserted into the DOM tree. Firefox saves the web page as represented by the DOM tree and not the original sent HTML source which results in saving not only the

[3] http://www.altavista.com

Example 4.1 The JavaScript double execution problem (from [112])

```
<script type="text/javascript">
   function WriteHello () {
     document.write('Hello ');
   }
   WriteHello();
   </script>
   <p>End</p>
```

javascript code but also the newly inserted "Hello" text tag. When the file is reopened the JavaScript code is executed again giving two "Hello" entries and so forth.

With the Expedia page in figure 4.2 this behaviour results in two menus and table row doubling which would be a serious problem (e.g. expecially for the table extraction domain). In contrast the Internet Explorer saves the original HTML code sent by the server without the JavaScript code executed. This avoids the JavaScript Double Execution problem but fails when the JavaScript code depends on an active internet connection for inserting dynamic content like the travel agency logos in the Expedia case.

4.3 WebPageDump Solution

The solution proposed by Pollak and Gatterbauer [112] was to make a visual local copy of the presented web page inside the browser. When not generating a ground truth but only a test database it is only necessary to provide information about the browser product itself. But this is secondary because WebPageDump is a Mozilla/Firefox extension that is only usable within the Mozilla/Firefox browser. While not being the commonly used browser, it has the advantage of its easy extension plugin concept. Also the existence of a ready made extension named Scrapbook [4] which tries to make better copies than the actual internal browser save function was an advantage because we used Scrapbook as base for our WebPageDump plugin. Whereas Scrapbook tries to generate a visual usable copy, the WebPageDump focus was the visual exact copy. Therefore we examined the problem of a measurement for the visual exactness of local copy and extended the Scrapbook approach according to the results (e.g. better Font/Unicode handling, addressing rendering bugs etc.).

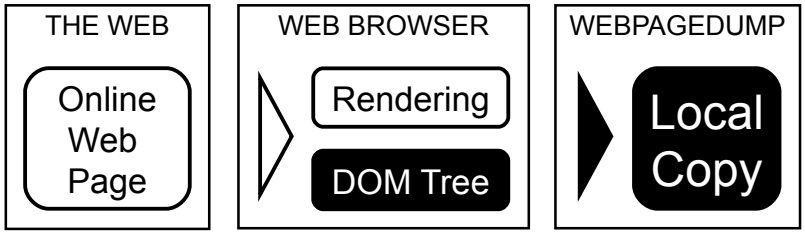


Figure 4.5: Concept of WebPageDump

For a better handling we introduced the WPD naming scheme resulting in easy readable short directory names by adding up the ASCII codes of the full URL (including GET variables) and applying a modulo 10,000 operation. At the end of the directory name the version is added (initially set to 0). If the same URL is stored twice this version counter is incremented indicating an additonal version (e.g. www_cnet_com_0003.1). In the case of an accidental identic WPD name but different URL an

[4] http://amb.vis.ne.jp/mozilla/scrapbook

additional counter is added seperated by a “c” (e.g. www_cnet_com_0003c1.0) so the basic scheme becomes

```
<domain_name>_<modulo_counter>[c<counter>].<version>
```

Figure 4.6 shows a sub selection of the WPD named directory listing from the Ventex test dataset. As one can observe, the naming has a good general readability compared to the use of the full URL (with the illegal characters replaced).

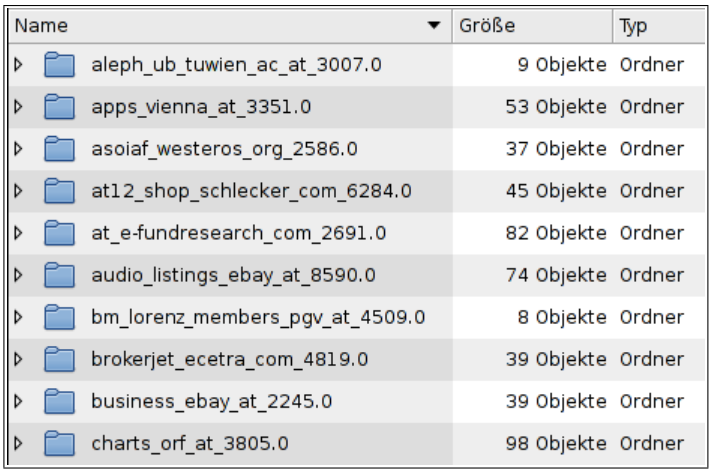


Figure 4.6: WPD naming example

5 The REDEVILA System

To analyse the possibilities of the visual web page analysis approach, we implemented a prototype system named REDEVILA (REcord DEtection on the VIsual LAyer) which is capable of finding blocks, doing the segmentation and classification of the segments, the importance ordering and the fine grained analysis inside the found segments based on the block structure. The system was implemented as a Mozilla/Firefox extension because we believe that the extension concept of this platform has much potential for the web page information extraction community in general (e.g. the LumberJaczk extension[1]) It would be easy for researchers to provide prototypes of their experimental systems and for others to test and evaluate these systems.

The system consists of various modules which provide the functionality needed. Figure 5.1 gives an overview over the basic architecture. Some small parts of the code base were originally developed for the VENTEX table extraction system. The experiments were executed and managed using additional extensions: the WebPageDump extension which was developed for generating a local web page repository (see chapter 4). The VTXServer extension that provides a telnet interface to the browser and due to its scripting functionality was very useful in executing mass tests (originally developed for the online test web interface of VENTEX). And the TinyAssert extension (based on the JavaScript assertion unit[2]) which provides an easy interface for executing tests inside the privileged Mozilla/Firefox environment.

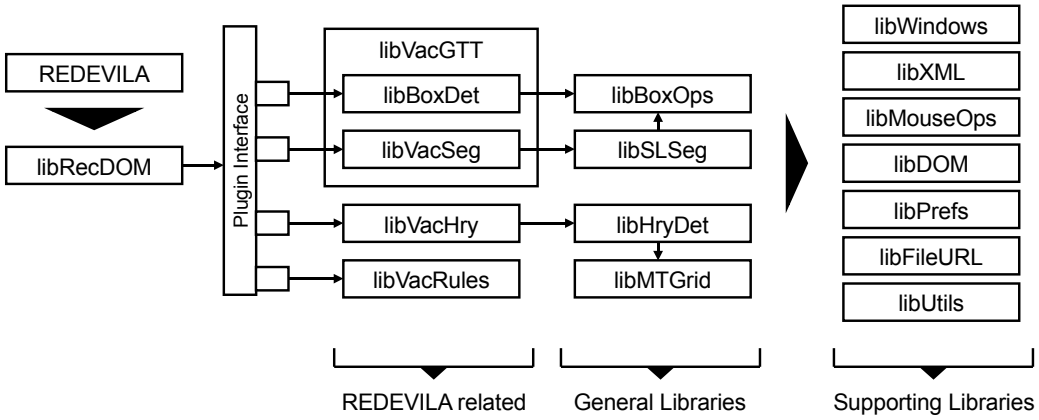


Figure 5.1: Basic REDEVILA architecture

5.1 Basic Model

To apply a specific extraction technology we have first to describe the kind of model on which the system is based. For this we will introduce the concept of *accentuation*. Accentuation draws attention by introducing an additional layer of semantics between the traditional word semantic

[1] http://lumberjaczk.org/
[2] http://jsassertunit.sourceforge.net/

and the geometric layout. Human speech for example uses variations in pitch, volume and also tempo to obtain accentuation. For our model we are observing two dimensional web pages which utilize dimensional and geometric compositions (layout) as well as font size and style (typography) for the accentuation.

(a) (b) (c)

Figure 5.2: Web page (a) with an accentuation (b) and a text stream only (c) version[3]

Figure 5.2 shows the difference between accentuation and word level (text stream only) semantic [48]. The accentuated version removes the world level semantics by unsharpening the picture. Similar to Doerman et.al. [45] we will introduce a third level between geometry and semantic called *functional level* which expresses this additional visual semantic level introduced by the concept of accentuation.

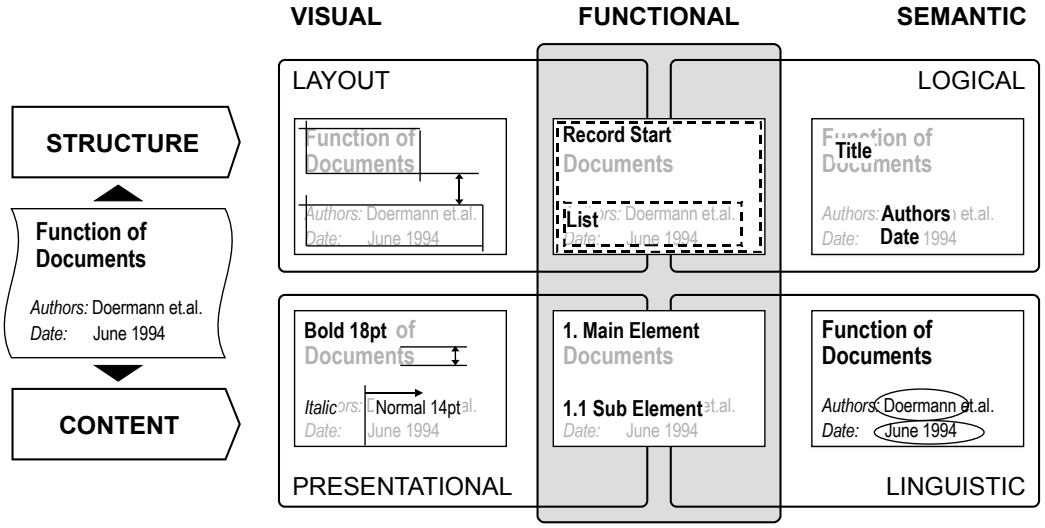


Figure 5.3: Geometric, functional and semantic descriptions (see also [45])

[3] http://www.ctv.ca

Summers [135] gives another but similar concept of a *visual logical structure* based on visual distinguishable segments which means that the logical structure analysis could be done without considering the semantics of specific words and gives the following definition:

> The logical structure of a document consists of a hierarchy of segments of the document, each of which corresponds to a visually distinguished semantic component of the document. Ancestry in the hierarchy corresponds to containment among document components [135]

The REDEVILA system could therefore be defined as a functional semantics analyzer by detecting general layout semantics which are based on the concept of accentuation. There exists a trade-off between domain-independence and logical/linguistic semantics. If we introduce more domain-dependence the focus shifts from the functional level to the semantic level but all based on the visual layer without considering the specific words. For example the appearance of a letter or a newspaper incorporate various domain-dependent visual semantics like usual positions for the title, the author, the date or the subject. Figure 5.4 shows a accentuated view of both a letter and a newspaper. It is interesting to discover the implied domain-dependent visual semantics without knowing the specific words. The letter seems to contain the address information at the top right and probably left above the salutation and the newspaper has a title, probably a short summary and the article text.

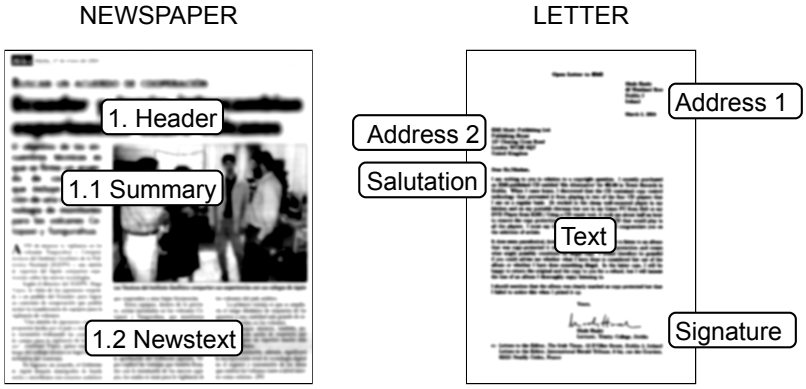


Figure 5.4: Domain dependent visual semantics of a newspaper and a letter[4]

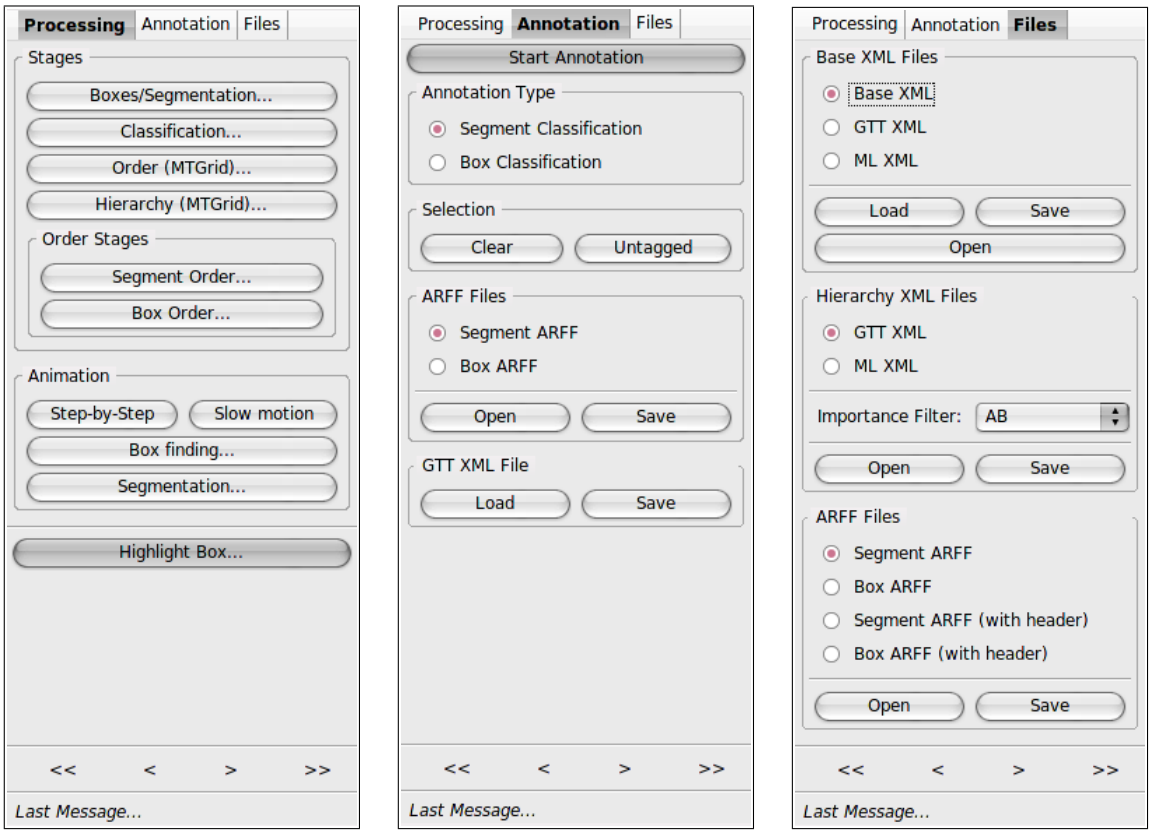
5.2 User Interface

The user interface is located in a special sidebar on the right side of the browser window with three tab sheets containing the controls for the processing, the annotation and the file handling (see figure 5.5)

5.2.1 Processing

The processing tab sheet (figure 5.5(a)) contains all controls that are relevant for the main processing stages. The four buttons at the top are used for the standard processing: (1) box identification and segmentation, (2) decision rule classification (3) order determination and (4) hierarchy detection. The controls below are utilized for a more fine grained control with the two stages of segment and

[4] http://www.bosai.go.jp/e/international, http://www.dsg.cs.tcd.ie/h̃aahrm/copying-protected-cds



(a) (b) (c)

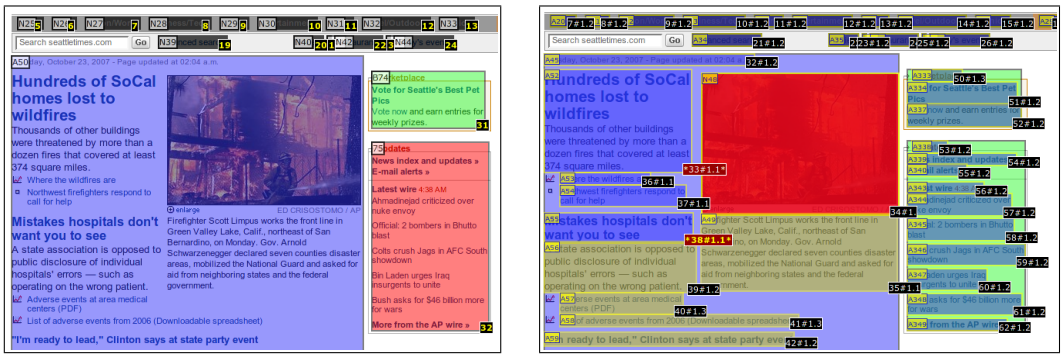
Figure 5.5: The REDEVILA user interface, (a) Processing (b) Annotation (c) Files

box hierarchy detection. The animation area is for the box identification and the segmentation and allows not only the animation but also step-by-step processing. Also included are two buttons for the separate handling of the box identification and the segmentation that enables a fast jump at the end of the corresponding stage. The last button at the bottom ist for highlighting a specific box after entering the box id.

5.2.2 Annotation

The annotation tabsheet (figure 5.5(b)) is for the ground truthing respectively annotation process. After selecting the annotation range (segment or box classification), the annotation can be started with the button at the top. The annotation itself is done by using the mouse and key controls (see table 5.1). Figure 5.6 shows a part of a web page[5] during the annotation process. The classification is reflected by various colors: blue for importance A, green for importance B, black for noisy blocks, light yellow for the selected boxes and red for the currently highlighted box. Boxes which are not annotated are shown in a more transparent red color. At the top left of each box is a small text field for the actual importance state together with the ID. The bottom right box shows the current order and hierarchy. For applying the key commands to a specified box, a selection or highlighting is required. Selection is done either by simple clicking of the desired box or using a rubber band for selecting more boxes at once. The selection could also be modified by the two selection buttons

[5] http://seattletimes.nwsource.com/html/home/index.html



(a) (b)

Figure 5.6: Annotation of a web page, (a) Segments (b) Boxes

Table 5.1: Key commands for the annotation

Operation	Key (segment)	Key (box)
clear block state	C	C
noisy block	N	N
importance	A, B	
increment order	X	X
decrement order	Y	Y
switch record level		R
switch new record flag		S
set hierarchy		1-9
deactivate block		D

directly from the tabsheet: “clear” simply clears any selection and “untagged” selects all boxes which are untagged (this is useful if there are very small boxes which are not possible to select). Two buttons for opening and saving the ARFF files which are used as the WEKA workbench input are placed below. The last two buttons at the bottom are for the XML ground truth file.

5.2.3 File Handling

The files tab sheet (figure 5.5(c)) puts together all the file handling including the four XML file types and the ARFF file output (with and without ARFF header). The buttons for the first three types are placed at the top whereas the hierarchy XML files allow more control over the importance set to be saved. At the bottom there are the ARFF file controls with two basic modes: (1) saving the files with an ARFF file header for direct processing and (2) saving the files without the header allowing the combination of different ARFF files for batch processing (the header would then be added seperately at the end). The following screenshots show a simple web page and the corresponding base XML output after the box identification and the segmentation on the right side. While there seems to be an inaccuracy in the box below the title, this is not the case. The text lines are not aligned because there is no line spacing information saved. Therefore only the box dimensions are present and filled up with the text by a standard line spacing. The same holds true for the box on the right.

5.3 Box Identification

An essential point of our algorithm is the identification of the box elements needed for the segmentation step. Unfortunately this is not a trivial task because the GECKO rendering engine hides these details in the XPCOM interface. There exists a Mozilla/Firefox specific function for getting the box coordinates from DOM nodes but this function does not consider complex line break situations resulting in false positions. Therefore we had to temporarily wrap each word with a custom tag which we called "X-Tag" (the reasons are discussed in detail in the file-format section) and calculate the dimension and position of the box out of the single word positions. Figure 5.7 shows the HTML code with the surrounding x-tags, a screenshot from the DOM tree and the resulting rendering together with the X-Tag based bounding boxes and the merged final single bounding box.

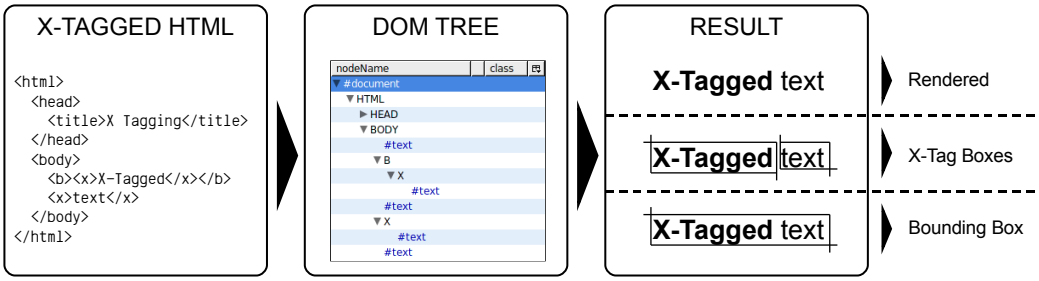


Figure 5.7: X–Tagging concept with rendered result, X–Tag based boxes and resulting bounding box at the bottom right

For the VENTEX system [59] we used the word positions directly which is more accurate than merging the positions of the words to a single rectangular box but needs much more disk space for the resulting XML file. The reason for developing the word box approach was to have a more general description of the web page also regarding other IE systems developed at our group and the development of a general table ground truth methodology. For the REDEVILA system we tried to make the boxes as large as possible for easier segmentation and therefore decided not to operate on the single word level. But this approach led to problems for the text indention and hierarchy detection which were solved by using an additional box parameter named ox, which gives the text indent in relative pixel width. Figure 5.8 from the online presence of the French newspaper Le Monde show some news boxes without (a) and with (b) ox attribute inside the segmentation output xml file.

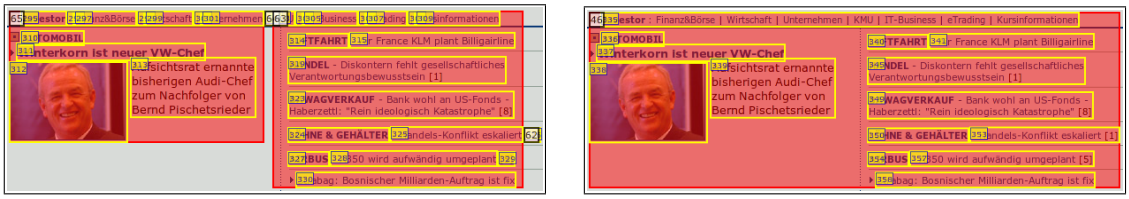
(a) (b)

Figure 5.8: Comparison, (a) without ox indention attribute (b) with ox indention attribute

One remaining difficulty is to consider the background because it might be generated by elements in the back. It would be possible but very inefficient to look up all elements for each single element to check if it is sitting in the background and obtaining the relevant properties from this element. For this case we depend on the DOM structure, handling the background information while traversing

through the DOM tree. This is, of course, not an APS algorithm but it is fortunately seldom that essential elements are positioned in such an absolute manner and the background is only one valuable attribute besides the font attributes. This problem could be solved by either modifying the source code of the rendering engine or probably by using an image processing algorithm which analyses an image of the whole web page for a positional color lookup. Recent Mozilla/Firefox versions have a "canvas" data structure which could be used for getting such an image of a web page. Another possible solution would be the generation of a 2D grid for a more efficient element look up.

Another problem is the use of absolute positioned DIVs for producing an overlay window effect. This was addressed manually by deleting the relevant code out of the locally saved web page because it would make no sense for the used algorithms. This is only valid for initial overlays during the page load not for user triggered overlays like menus.



(a) (b)

Figure 5.9: Comparison, (a) without box merging (b) with box merging

To improve the segmentation some boxes are merged together when specific conditions are met (Cleaning Algorithm). This merging is based on a DOM tree neighbourship relation for easier implementation and is therefore not an APS algorithm (but this could be overcome by ordering the boxes according to their position, which was not done for the reason of easier implementation). The DOM neighboured boxes are investigated according to a containing or neighbourship relation. Without this, e.g. horizontal aligned links would normally be identified as single blocks (see figure 5.9). Of course there is also a minimum width threshold for vertical separators which would probably lead to the same segmentation result but with the boxes themselves remaining separated and therefore exceeding the complexity of the ordering and hierarchy analysis.

5.4 Segmentation

Because of the sometimes complex visual structure it is mostly not possible to analyse the visual attributes directly on the whole web page. The meaning of font attributes often depend on the position inside a visual block and the whole web page. For example the same larger font could be used as a header and also inside the text and only the top position marks the specific text as a header.

Moreover the web page layout has to be considered to exclude unimportant text. Of course the decision of what is important is difficult and is based on various assumptions which will be discussed later. Therefore we have to segment the web page for extracting the larger visual blocks at first. There are different segmentation algorithms available as described in chapter 2.

We are applying the VIPS approach by projecting the boxes into a plane as large as the web page and divide the plane successively until only the separator lines are left. This is an incremental process that has to be repeated several times until an ending condition is reached. The main advantage is the possibility to consider features directly from the projected blocks in order to improve the visual exactness of the segmentation process which would be difficult, for example, with projection profiles.

The algorithm works as follows: after finding the various boxes we create two planes, one for the horizontal and one for the vertical separators, each having the size of the whole web page. In the case of frames we treat each frame as a single web page. The segmentation process is done by projecting each block from the block list into the initial large separators and apply the basic block operations based on the overlapping situation (see figure 5.10(a)). The resulting segments are calculated by the inverted cut-set out from the remaining separators (see figure 5.10(b)). The inversion results in various segments which are processed again until the ending condition of one remaining separator is reached (which has shown a good segmentation performance).

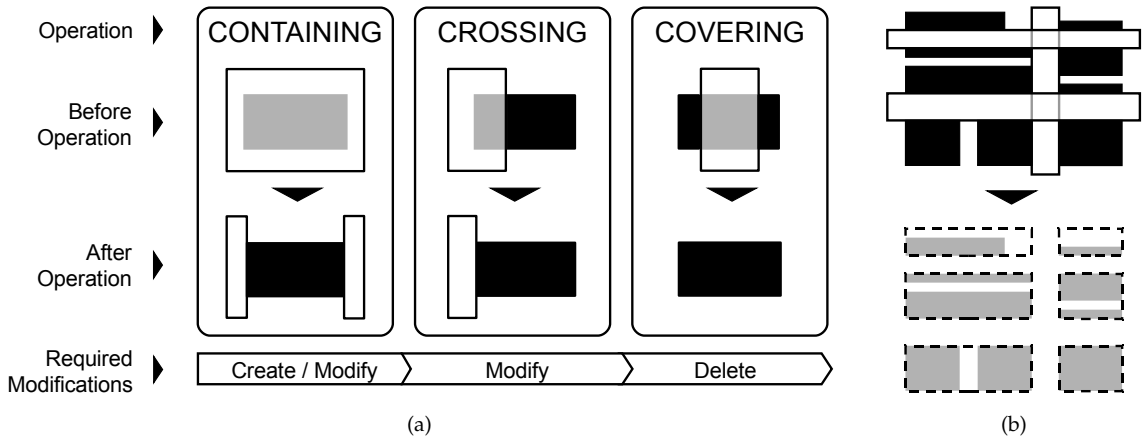


Figure 5.10: (a) Basic block operations for the vertical separator case (b) Separator invertion for the segment generation

In contrast to the VIPS approach we are interested in a fine grained segmentation in the sense that we want to find all visual blocks separated by whitespace (beside various conditions like the minimal width or height). So we have no excplicit stop condition except we don't find any more white space.

We define a segment as a minimal rectangle around a set of one ore more blocks. The segmentation process itself is described as follows (see also [28] and [27]):

A web page is defined as a triple by

$$\Omega = (O, S, \varrho) \tag{5.1}$$

with

$$\begin{aligned} O &= \{o_1, o_2, o_3, \dots\} \\ S &= \{s_1, s_2, s_3, \dots\} \\ \varrho &= O \times O \rightarrow S \cup \{\} \end{aligned} \tag{5.2}$$

whereas O describes a set of objects respectively sub web pages, S is a finite set of separators and the relation ϱ is the mapping of the object set cross product into the separators.

Each horizontal separator has additional properties which are set according to the projected boxes. These additional properties are a threshold, the background color and the font height for both the top and the bottom of the separator. Top and bottom is related to the projected box, therefore top means that the separator properties are modified by the box below and reverse. The vertical separators only have the general threshold for both the left and the right borders.

Algorithm 5.1 Bottom deletion algorithm for a horizontal separator

Input: *hSep*: horizontal separator; *vBox*: current projected VEN box
Return: ∅

```
1:  function DELETEHORIZSEPARATOR(hSep, vBox)
2:    if hSep.height < hSep.bottom.threshold then
3:      if vBox = text then
4:        if (hSep.bottom.threshold ≥ hSep.top.threshold) ∨
5:          (hSep.bottom.fontHeight/2 ≥ hSep.top.threshold) ∨
6:          (hSep.bottom.fontHeight/2 > hSep.height ∧
7:          hSep.bottom.bgColor = hSep.top.bgColor) then
8:            DELETESEPARATOR(hSep)
9:        EndIf
10:     else
11:       DELETESEPARATOR(hSep)
12:     EndIf
13:   EndIf
14: end function
```

This properties determine at every projection stage if the separator should be deleted or not. The rules are described by algorithm 5.1

In the case of a text box the horizontal bottom threshold is set to the font height otherwise a constant threshold of 7 pixels is used for *horizontal.bottom* and *vertical.right*. The first line checks in general if the height of the separator is smaller than the bottom threshold. If this is the case the separator is a candidate for deletion assuming the following rules are met: (1) For a text box line 4 checks if the font of the box above is greater or equal to the box below, (2) line 5 checks if the height of the separator is smaller than the half font size or, in other words, boxes are near enough and (3) line 6 together with line 7 handles the case of a greater distance (but below the threshold due to line 2) than half of the font size but similar background color. Figure 5.11 illustrates these cases.

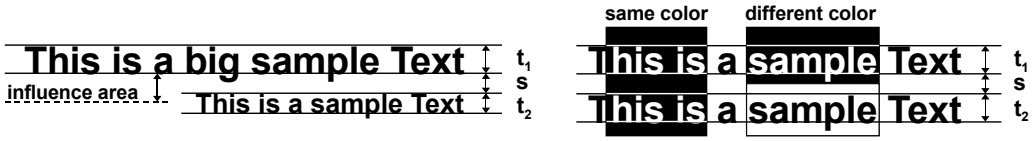
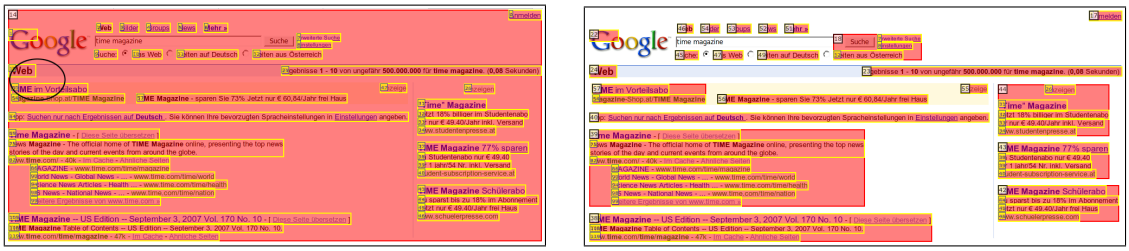


Figure 5.11: Bottom font size dependent deletion rules

Figure 5.12 shows a comparison of a Google search result[6] regarding the third rule. Figure 5.12(a) is much less separated because of the equal font size between the "Web" text at the top of the page and the text below (the area in the ellipse). Figure 5.12(b) separates much better due to the consideration of the different background colors.

The segmentation algorithm itself is applied by a top-down approach with a horizontal and vertical separator of the size of the whole web page as a start. In contrast to the VIPS algorithm with its degree of coherence (DOC) measure, we do not rely much on the DOM tree for the block extraction process, instead we include some of the decisions into the segmentation stage (which is by VIPS referred as separator detection stage). Therefore our approach gains more DOM tree independence resulting in a pure APS algorithm. Would it be possible to include also a kind of degree of coherence in a DOM tree independent APS manner? We think yes – by changing the ending condition which reflects the remaining separator count. Figure 5.13 depicts the segmentation result with an ending condition of 3 instead of 1. Of course, it is not enough to set an absolute value. We would need an

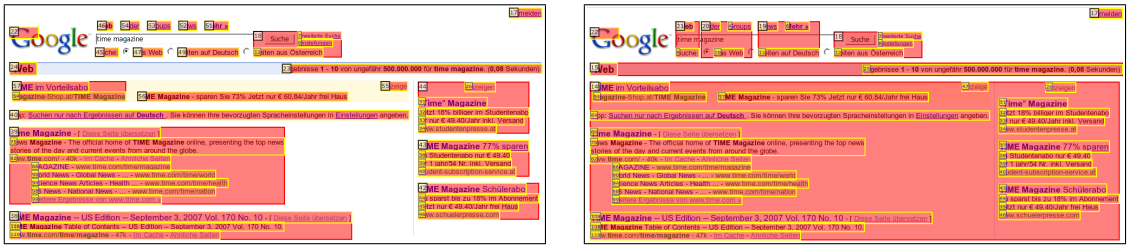
[6] www.google.at



(a) (b)

Figure 5.12: Comparison, (a) without color correction (b) with color correction

additional measure with dynamic adaption, probably dependent from the web page area and the block density.



(a) (b)

Figure 5.13: Comparison, (a) standard segmentation and (b) different ending condition (3 separators)

Figure 5.14 illustrates the full segmentation process (for this the animation/step-by-step feature of the REDEVILA system was used)

Figure 5.14(a)–(c) shows the initial horizontal and vertical separators already splitted by the first projected box on the left and the continued segmentation on the right. Figure 5.14(c) shows the first segmentation result with the found segments marked by the dashed rectangles. Note that the first box from the center segment is already projected. Figure 5.14(d) shows the box projection inside the last processed segment and figure 5.14(e) the final segmentation result.

The general algorithm for the segmentation process is described in algorithm 5.2. The already projected boxes are collected inside the segment where they are matched (see also line 10), resulting in a much better performance because at each segmentation step only the boxes inside the parent segment have to be considered. Also some optimizations are applied which utilize the fact that the boxes in the list have the same ordering like inside the DOM tree where spatial related boxes often reside after each other. This is not a necessary condition for the algorithm but improves the performance if it is the case (these optimizations are not reflected in the algorithmic description).

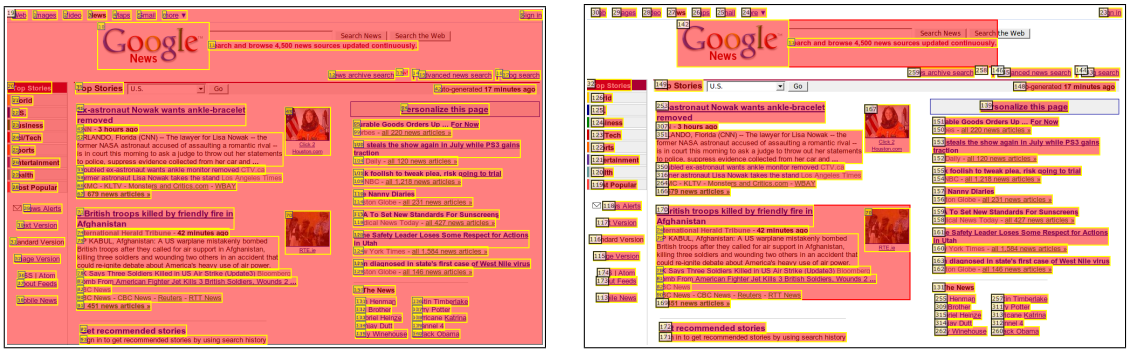
Unfortunately the segmentation approach has also some limits when applied to so called “L-Shapes” [100] which are layouts that do not allow a horizontal or vertical separator over the whole width or height of the parent element. Figure 5.15 shows another case where the segmentation is not optimal. The various elements at (a) are very near together and the heuristic separator deletion rules are too ”strong”. For comparison, (b) shows the result without any rules applied. Though giving a segmentation, the segmentation is much too fine grained.

(a) (b)

(c) (d)

(e) (f)

Figure 5.14: Step-by-step segmentation process



(a) (b)

Figure 5.15: Comparison, (a) with rules (b) without rules

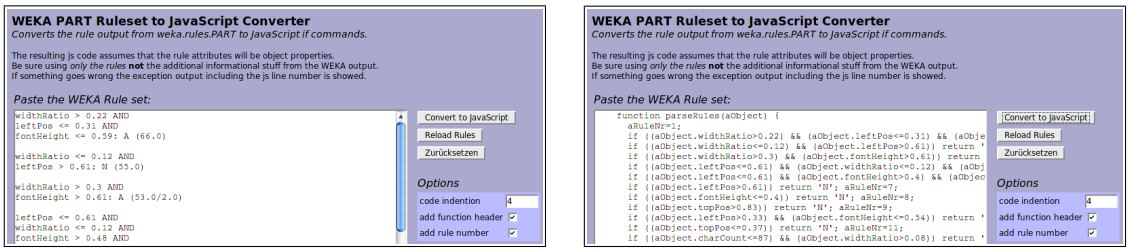
Algorithm 5.2 REDEVILA segmentation algorithm

Input: *vBoxList*: set of VEN boxes; *pageWidth*, *pageHeight*: page dimensions
Return: *pSegList*: set of resulting segments

```
1:  function SEGMENT(vBoxList, pageWidth, pageHeight)
2:    pSegList ← INITSEGMENT(pageWidth, pageHeight)
3:    pSegList.vBoxList ← vBoxList
4:    while ∃pSeg ∈ pSegList with ¬pSeg.closed do
5:      foreach pSeg ∈ pSegList with ¬Seg.closed do
6:        foreach vBox ∈ pSeg.vBoxList do
7:          foreach cSeg ∈ pSeg.cSegList do
8:            if CONTAINS(cSeg, vBox) then
9:              PROCESS(cSeg, vBox)
10:             cSeg.vBoxList ← vBox
11:           EndIf
12:         end for
13:       end for
14:       foreach cSeg ∈ pSeg.cSegList do
15:         cSeg.cSegList ← INVERTSEPARATORS(cSeg)
16:         cSeg.closed ← CLOSECONDITION(cSeg)
17:         pSegList ← pSegList ∪ cSeg
18:       end for
19:       pSegList ← pSegList \ pSeg
20:     end for
21:   end while
22:   return pSegList
23: end function
```

5.5 Importance Classification

For the importance classification of the segments we applied a decision rule algorithm using he PARTS algorithm from the WEKA workbench [144] based on the C4.5 decision tree learning algorithm [118] which is a successor of the ID3 (Iterative Dichotomiser 3) algorithm [119] (see also algorithm 5.3). We selected the PART algorithm because of its simplicity, overall good performance and because of the easy integration into the JavaScript code by converting the WEKA output directly into a set of IF THEN commands through a simple also JavaScript based HTML interface (figure 5.16).



(a) (b)

Figure 5.16: The HTML PART Rules to JavaScript converter, before (a) and after (b) the conversion

The ID3 algorithm is based on choosing the ”best” classifier for the successive divide and conquer principle of the decision tree construction. This decision is based on the concept of entropy ex-

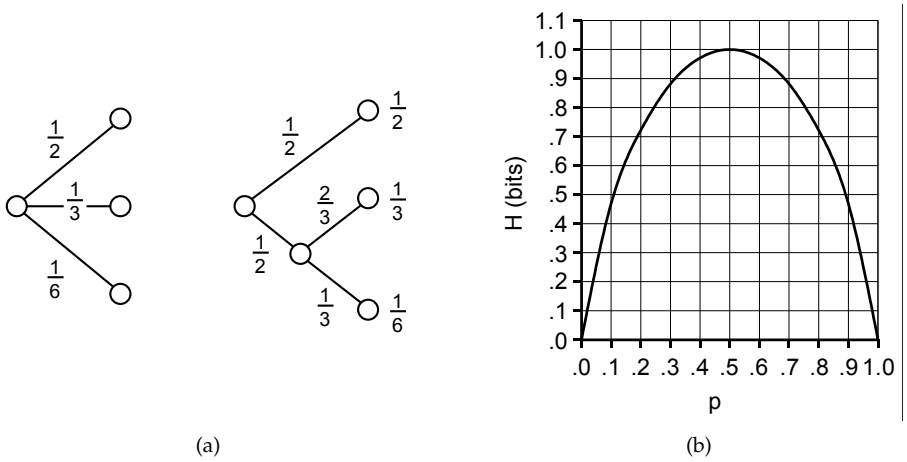


Figure 5.17: (a) Choice decomposition (b) Entropy of two possibilities; (after [126])

pressing the amount of "choice" which is involved in the selection of a specific attribute (see figure 5.17(a)). Mitchell [102] gives a detailed introduction into decision trees with entropy defined as

$$Entropy(S) \equiv \sum_{i=1}^{c} - p_i \, log_2 \, p_i \tag{5.3}$$

S is the example set, c the count of different possible values for the target attribute and p_i the part of S belonging to a class i. Based on this entropy definition the information gain, as a measure for the effectiveness in classifying the set S with an Attribute A, is introduced (see formula 5.4) and forms the base for the ID3 algorithm.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5.4}$$

While impressive in its simplicity the main disadvantage of ID3 is its tendency to overfit the training data. A hypothesis is said to overfit the training data if there exists another hypothesis with less accuracy on the training data but more precision on the entire distribution. The C4.5 algorithm uses rule post-pruning for avoiding overfitting where the resulting decision tree is converted to an equivalent set of rules. The resulting rules are then pruned (generalized) by removing preconditions and sorted according to their estimated accuracy [102].

The WEKA PART algorithm generates decision rules similar to the C4.5 method but instead of using the complete set it is based on the remaining set of examples building partial trees and tries to find the most general rule by selecting the path to a leaf that covers the greatest number of instances [144].

The selection of valuable and reasonable features is not an easy task and there exists no single state of the art algorithm for determining the "best" features (see also [115]). Because of our visual related extraction algorithm we selected various visual related features and normalized them with maximal values regarding the whole web page. Figure 5.18 shows all initial features with their importance distribution and table 5.2 describes the inital features in detail. To reduce the attribute count we applied the CfsSubsetEval evaluator together with a simple BestFirst Search. The resulting final features are placed at the top of table 5.2 with a mark inside the "selected column". There is nearly no effect on the error rates which reflects the redundancy of the initial feature set.

Algorithm 5.3 The ID3 decision tree induction (after [102])

Input: *Examples*: training examples; *TargetAttribute*: attribute which should be predicted; *Attributes*: other possible attributes
Return: *finalID3decisiontree*

function BUILDID3TREE(*Examples*, *TargetAttribute*, *Attributes*)
  root $\leftarrow \varnothing$
  if $\forall$*Examples* $= \oplus$ **then**
    Return (*root.label* $\leftarrow \oplus$)
  else if $\forall$*Examples* $= \ominus$ **then**
    Return (*root.label* $\leftarrow \ominus$)
  else if *Attributes* $= \varnothing$ **then**
    Return (*root.label* $\leftarrow$ MOSTCOMMON(*TargetAttribute*))
  else
    $A \in$ *Attributes* with HIGHESTINFGAIN(*Attributes*, *Examples*)
    root.decision $\leftarrow A$
    foreach $v_i \in A$ **do**
      treeBranch $=$ NEWBRANCHBELOW(*root*) corresponding to $A = v_i$
      $Examples_{v_i} \subset Examples$ with $Examples_{v_i}[A] = v_i$
      if $Examples_{v_i} = \varnothing$ **then**
        node $\leftarrow$ ADDLEAFNODE(*treeBranch*)
        node.label $=$ MOSTCOMMONATTR(*TargetAttribute*, *Examples*)
      else
        subtree $\leftarrow$ buildID3Tree($Examples_{v_i}$, *TargetAttribute*, *Attributes* $- \{A\}$)
        node $\leftarrow$ ADDSUBTREE(*treeBranch*)
      EndIf
    end for
  EndIf
end function

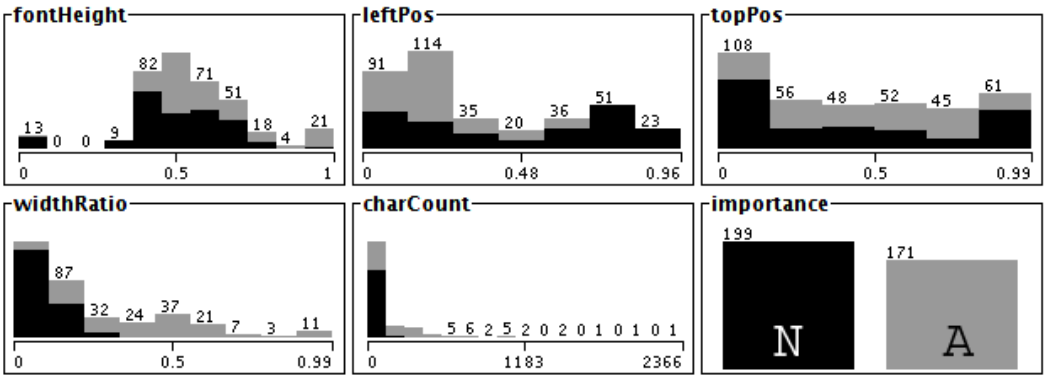


Figure 5.18: The final feature set and the corresponding importance A and N (noisy block) distribution

Table 5.2: All initial and selected (order top to bottom) features

Attribute	Selected	Description
leftPos	×	left position in relation to the web page width
topPos	×	top position in relation to the web page height
widthRatio	×	width in relation to web page width
charCount	×	count of single chars (without whitespace)
wordCount	×	count of words
wordLinkCount	×	count of words which represent a link or are part of a link
fontHeight		font height in relation to maximum web page font height
fontWeight		font width in relation to maximum web page font width
heightRatio		height in relation to web page height
areaRatio		area in relation to web page area
linkCount		count of links
objectCount		count of objects (text blocks and images)
wordLinkRatio		link word count in relation to word count (wordLinkCount/wordCount
linkObjRatio		link word count in relation to object count (linkCount/xCount)
importance		N = noisy blocks, AB = importance with A representing ”more” importance

Table 5.3: Detailed accuracy by class

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
		ABN Setup			
0.895	0.101	0.884	0.895	0.890	A
0.714	0.067	0.656	0.714	0.684	B
0.804	0.093	0.846	0.804	0.824	N
		AN Setup			
0.912	0.095	0.891	0.912	0.902	A
0.905	0.088	0.923	0.905	0.914	N

For the performance measure the standard 10 fold cross-validation was applied. As table 5.3 shows the class B has the largest errors which reflects the principal problem of differing between noisy and less important blocks. To overcome this problem and to improve the accuracy of the classification result we decided to merge the B importance class together with the noisy block (N) class. Also because we are interested primarily in the main content which is expressed by the importance class A only.

Table 5.3 compares the accuracy measures for the two different feature sets with the three-class solution at the top and the two-class solution below. The correctly classified instances were improved from initial 83% up to 91%. Table 5.4 shows the details and an additional comparison with the ZeroR classifier which simply sets all classes to true. Figure 5.18 shows the final feature set with their importance distribution and appendix A gives the detailed WEKA classification outputs.

Table 5.4: Comparison between ABN, AN and the ZeroR classifier

Setup	Correct	Incorrect
ABN	309 (83.51%)	61 (16.49%)
AN	336 (90.81%)	34 (9.19%)
ZeroR	171 (46.22%)	199 (53.78%)

Not considered by REDEVILA but interesting for visual web page features in general was an observation made by Song, Liu, Wen and Ma [131] regarding the height normalization of very large web pages. If the web page height is many times the height of a standard resolution the classification of important blocks could fail because their features become probably similar to the features of unimportant noisy blocks (e.g. advertisement). They propose window spatial features by not using a relative proportional height but a fixed value (see formula 5.5). Also the BlockCenterY feature is modified according to formula 5.6

$$BlockRectHeight = \frac{BlockRectHeight}{WindowHeight} \tag{5.5}$$

$$BlockCenterY = \begin{cases} \dfrac{BlockCenterY}{2 \cdot HeaderHeight} & \text{if } \boldsymbol{BlockCenterY} < HeaderHeight \\ 0.5 & \text{if } HeaderHeight < \boldsymbol{BlockCenterY} \ldots \\ & \quad \ldots < PageHeight - FooterHeight \\ \dfrac{1 - (PageHeight - BlockCenterY)}{2 \cdot FooterHeight} & \text{otherwise} \end{cases} \tag{5.6}$$

5.6 Multitopological Grid

The multitopological grid is based on a minimal grid with a logical coordinate system. The main idea was to develop a simple and efficient data structure for applying spatial reasoning by the modeling of rules.

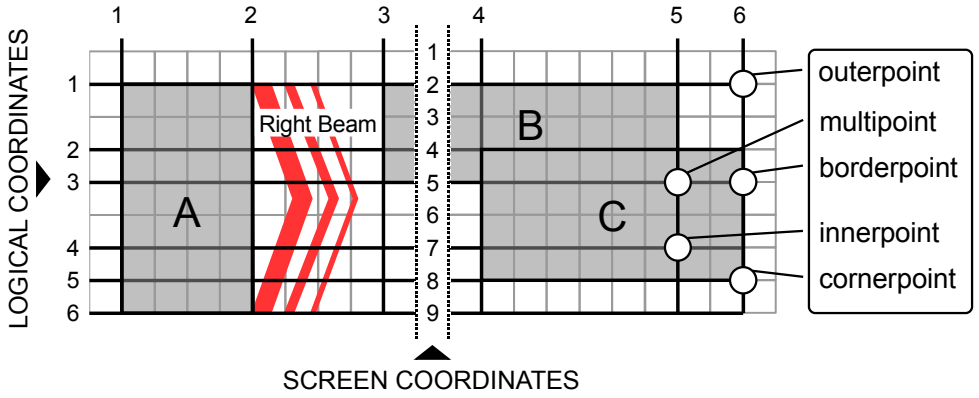


Figure 5.19: Spatial Reasoning Grid with logical/screen coordinates, point types and an example right beam

Spatial relations could be generally described by three different basic semantics [47]: (1) metric (e.g distance, fontHeight), (2) topological (e.g. within, overlap) and (3) directional (e.g. right, below). While not in the narrow sense beeing a topological data structure the term "topological" is used in a wider sense meaning support for the ordering analysis of the elements inside the grid whereas the term "multi" expresses the ability to provide basic support for the reasoning with all three spatial semantics. The REDEVILA system splits the reasoning step into two different specialized stages to simplify the reasoning process: (1) the ordering of the spatial objects and (2) the hierarchy analysis (see section 5.7 and 5.8).

The multitopological grid (MT Grid) is based on the visual rendering result provided by a browser following the CSS 2 box and formating model [19]. Similar to our previous double topological grid approach we refer to such rendered rectangles as visual element nodes (VEN). VENs are represented by a visual box (VB) containing either a single VEN or multiple VENs based on the bounding box (see also figure 5.7).

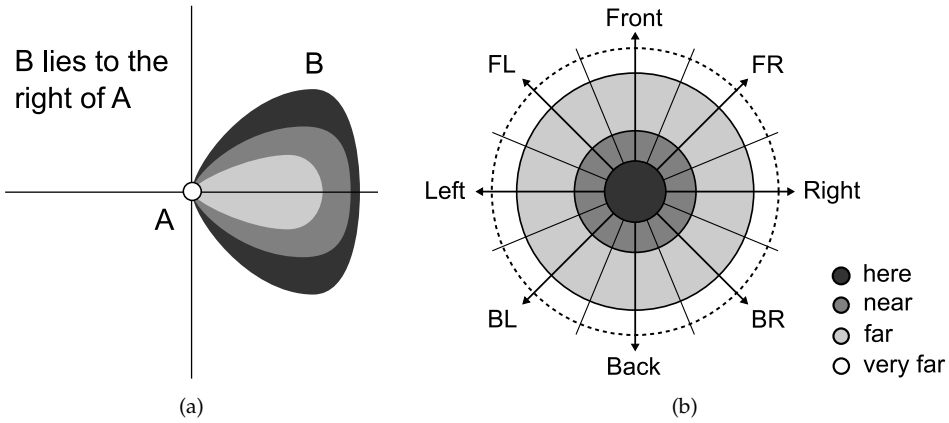


Figure 5.20: (a) Ratings of acceptability (after [80] fig.1), (b) Qualitative egocentric distances and directions (after [55] fig.6)

A web page is described by a visual topological box model $V = \langle \mathcal{B}, \mathcal{M}_b \rangle$. In contrast to the VENTEX system we don't consider single word boxes. Each visual box $\mathbf{b} \in \mathcal{B}$ consists itself of two vectors $\mathbf{b} = \langle \mathbf{c}, \mathbf{a} \rangle$ with $\mathbf{c} = \langle x^b, y^b, w^b, h^b \rangle$ as the upper-left coordinate with the box dimensions and $\mathbf{a} = \langle a_1, ..., a_n \rangle$ as property-value pairs providing some additional attributes like typographic information. In the case of multiple VENs the attribute list is built by either using the maximum values (e.g. fontHeight, fontWeight) or by summing up counting properties (e.g. wordCount, LinkCount). This set of visual boxes $\mathcal{B}$ is represented by a minimal grid data structure which we will call the multitopological grid $\mathcal{M}_b$ and is basically defined by

$$\mathcal{M}_b = \{ \langle g,\, P,\, B \rangle \mid g \in \mathcal{G},\, P \subseteq \mathcal{P},\, B \subseteq \mathcal{B} \cup \emptyset \} \tag{5.7}$$

whereas g is the logical grid coordinate of the minimal grid structure $\mathcal{G}$ (5.8), P defines the grid point type based on the set $\mathcal{P}$ (5.9) of available types and B are the boxes at the specific grid point with the empty set as additional "box" type for outer points.

$$\mathcal{G} = \{ \langle x^g, y^g \rangle \mid x^g = f_x(x_{i_x}), y^g = f_y(y_{i_y}) \} \tag{5.8}$$

$$\mathcal{P} = \{ cornerpoint, innerpoint, outerpoint, borderpoint, multipoint \} \tag{5.9}$$

$f_x : x_{i_x} \mapsto i_x$ and $f_y : y_{i_y} \mapsto i_y$ are the logical mapping functions which map the physical screen coordinates to the minimal logical visual box coordinates based on the following sorted vectors by indexing the physical screen coordinates of the left and right borders of the visual boxes.

$$\begin{aligned} \mathbf{x} = \langle x_1, \ldots, x_{v_x} \rangle \text{ with } & (x_{i_x} = x^b \vee x_{i_x} = x^b + w^b) \wedge (x_{i_x} < x_{j_x}) \\ & \forall j_x > i_x,\ 1 \leqslant i_x \leqslant v_x,\ 1 \leqslant j_x < v_x,\ v_x \leqslant |\mathcal{B}| \\ \mathbf{y} = \langle y_1, \ldots, y_{v_y} \rangle \text{ with } & (y_{i_y} = y^b \vee y_{i_y} = y^b + w^b) \wedge (y_{i_y} < y_{j_y}) \\ & \forall j_y > i_y,\ 1 \leqslant i_y \leqslant v_y,\ 1 \leqslant j_y < v_y,\ v_y \leqslant |\mathcal{B}| \end{aligned} \tag{5.10}$$

As a complement we introduce the inverse mapping functions $f_x^{-1} : i_x \mapsto x_{i_x}$ and $f_y^{-1} : i_y \mapsto y_{i_y}$ for converting logical coordinates back to screen coordinates.

Figure 5.20(b) shows principal directional and metric relations based on an egocentric system [55] which is the operation principle used by the REDEVILA system. Through the splitting of the recognition process into an ordering and hierarchy analysis stage every element is analyzed one after the other by an "egocentric" view.

Algorithm 5.4 Multitopological grid generation

Input: *boxList*: set of VEN boxes;
Return: $\varnothing$

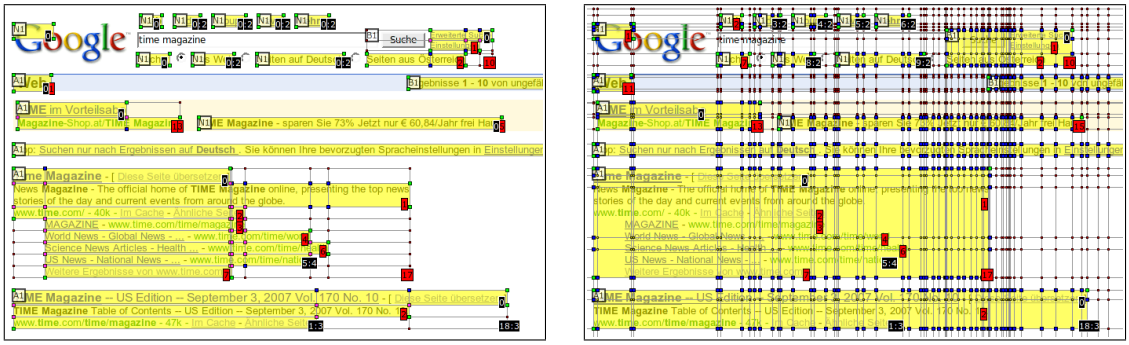
```
1:  function BUILDMTGRID(boxList)
2:      coord2idx ← []
3:      idx2coord ← []
4:      foreach box ∈ boxList do
5:          coord2idx[box.coordinates] ← 0
6:      end for
7:      foreach c ∈ coord2idx do
8:          idx2coord.push(c)
9:      end for
10:     foreach i ∈ idx2coord do
11:         coord2idx[idx2coord[i]] ← i
12:     end for
13:     MTGrid ← []
14:     foreach box ∈ boxList do
15:         MTGrid[coord2idx[box.coordinates]].box ← box
16:         MTGrid[coord2idx[box.coordinates]].type ← pointType
17:     end for
18: end function
```

One remaining problem with metric relations is the representation of distance, e.g. for which number n is an object "far" from the first object of a given sequence of n objects that are "close" to each other (see also [121] and figure 5.20(a)). Because we are targeted at web pages, we use the fontHeight as the base measure which is reasonable due to various topographical conventions. But for every rule (or habit) there exists an exception (or to say creativity) which limits our selected fontHeight approach.

For improving the scanning algorithms we defined 16 different bit constants for the point types as shown in table 5.5 with the basic bits set at the bit position for each type. The combination types have additional bits set according to their properties. For example CornerTopLeft (CTL) includes not only the bit for the type itself but also bits for the Corner (C), the left position (L), the top position (T) and the Border(B) because we define a corner as a subtype from border. Multiple points are detected by multiple boxes in the $\mathcal{B}$ set and described similar to single points by simply adding the point type bits. Algorithm 5.4 describes the basic grid bulding process.

Table 5.5: Bit settings and combinations for the various grid point types

Type	Abbr.	BitPos	CTL	CTR	CBL	CBR	BL	BR	BT	BB
Left	L	1	X		X		X			
Top	T	2	X	X					X	
Right	R	3		X		X		X		
Bottom	B	4			X	X				X
Corner	C	5	X	X	X					
CornerTopLeft	CTL	6	X							
CornerTopRight	CTR	7		X						
CornerBottomLeft	CBL	8			X					
CornerBottomRight	CBR	9				X				
Border	B	10	X	X	X	X	X	X	X	X
BorderLeft	BL	11					X			
BorderRight	BR	12						X		
BorderTop	BT	13							X	
BorderBottom	BB	14								X
Inner	I	15								
Outer	O	16								



(a) (b)

Figure 5.21: Sample multitopological grid applied to a web page for boxes (a) and segments (b)

Figure 5.21 shows a visualization of multitoplogical grids applied at box and segment level. The different point types (cornerpoint, innerpoint, outerpoint, borderpoint, multipoint) are colored differently. There are also some informational text boxes for the hierarchy and order analysis.

5.7 Ordering

Simple ordering algorithms order text blocks either left/top or top/left. As figure 5.20(a) shows, directional rules are ambiguous but the diagonal ordering approach tries to give a reasonable limit where the ordering direction should be changed. In contrast to the soft ordering algorithm from Mitchell and Yan [101], our diagonal ordering is targeted at the width of the blocks as required for our fine grained block ordering step. The various blocks are ordered based on a diagonal comparison. The algorithm uses the maximum width of the current main structure and the maximum width of the two compared blocks and is therefore a monotone algorithm which is of course the precondition for a sorting algorithm.

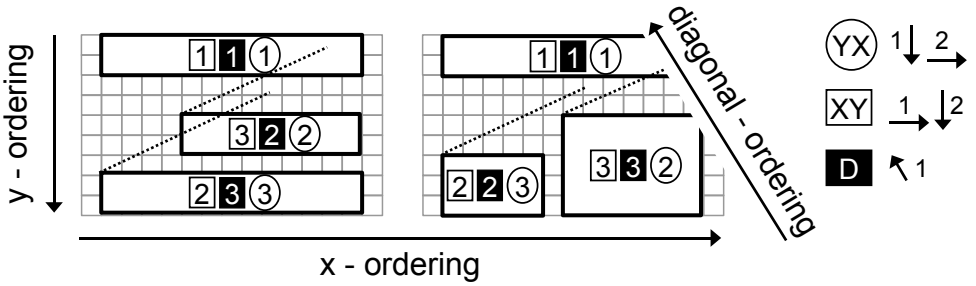


Figure 5.22: Comparison between simple X-Y, Y-X ordering and diagonal ordering

Figure 5.22 shows a comparison between the simple X-Y (left/top) and Y-X (top/left) ordering and the correct diagonal ordering. With the X-Y (white rectangle) and Y-X (white circle) ordering only one of the two different situations can be handled properly whereas the diagonal ordering approach (black rectangle) gives the intuitive "right" ordering.

The basic diagonal ordering formula is defined as follows

$$limit = \frac{1}{2 + \frac{2\,b_{max}}{w_{max}} \sqrt{\frac{b_{max}}{w_{max}}}} \tag{5.11}$$

b_{max} is the maximum width of the two compared boxes and w_{max} the maximum width of the main parent structure (segment or webpage). The limit is compared to the arcus tangens between the upper left corners of the two boxes. If the result is smaller than the limit we will sort left/top and top/left otherwise. Figure 5.23 shows different plots with varying maximum box widths (3,5,7 and 9) and the coordinates which are below (black) or above (white) the limit based on a maximum w_{max} of 10. The function will increase the slope of the limit if the maximum box width gets smaller. This reflects the principle that a smaller box has also a lower influence at the above/right area. With increasing box width the slope gets more flat and the ordering comes close to a simple left/top ordering.

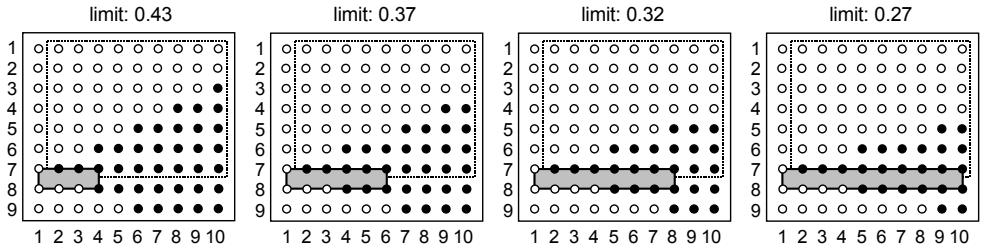


Figure 5.23: Various diagonal ordering plots according to the block width

Figure 5.24 gives an impression of the limits behaviour of the diagonal ordering function depending on different maximum box and page widths.

As mentioned before the multi-topological grid is used for extending the ordering precision. Example 5.1 shows a sample log output of the ordering process. We can see the result of the diagonal ordered search which returns the first undefined box (see also figure 5.21 for the detection principle). Afterwards two rules are checked and fulfilled which results in setting the order. RULE 1 is an alignment rule and RULE 6 is a distance measure.

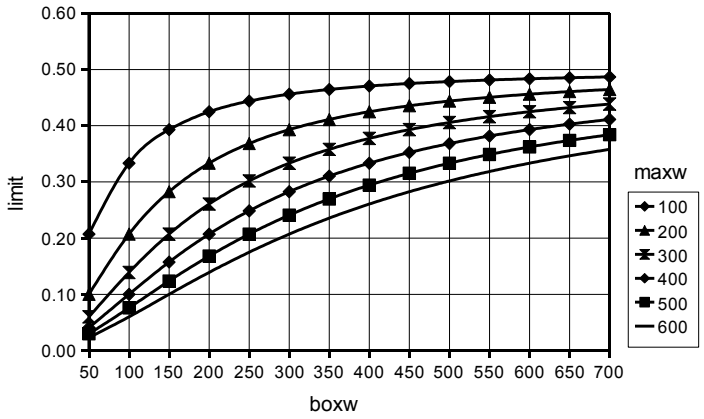


Figure 5.24: Limits of the diagonal ordering with different maximum box widths and page widths

Example 5.1 Sample log output from the ordering process

```
*libHryDetClass* processOrder.SETORDER ...
  id:56; order:13
*libHryDetClass* processOrder.processBoxes ...
  id:53; order:12
going through bottom boxes...
  FIRST UNDEFINED => id:56
*libHryDetRSClass* rule ...
  RULE1 => YES ...
    aBox1.id:53 - aBox2.id:56 (aligned)
  RULE6 => YES ...
    aBox1.id:53 - aBox2.id:56;
    aBox1.fontHeight:14; aThreshold:14 (near)
=> YES
```

5.8 Hierarchy

Principally we can differ between two kinds of hierarchy: (1) monohierachy and (2) polyhierarchy structures. A monohierarchy is a structure where every item has exactly one defined position inside the hierarchy and the classification is a definite one-to-one relationship. Many physical objects are organized in this way because they exist only once and this physical grounding is therefore very intuitive for humans. The Dewey Decimal Classification (DDC) for libraries is a good example for a monohierarchie [44]. But the injective one-to-one relationship is at the same time the greatest disadvantage of monohierarchies because most real world objects have in fact a polyhierarchical dimension when they are classified. Strictly speaking a polyhierarchy is not a hierarchy but a directed acyclic graph. Facette classification or other multiset approaches are examples of such structures [7].

The REDEVILA system is a monohierarchical analysis system because we are primarily interested in record detection and separation on a physical two dimensional grid. Also the used visual semantic approach is only able to detect simple relations which could be easy expressed by a monohierarchy and there is no need to introduce a more complex structure. Of course this would be different if a domain-dependent visual analysis is used because there are more specific logical semantic tags like shown in figure 5.4.

We applied a hierarchy model with a maximum depth of two levels which should be adequate for most standard situations. Beside this we have implemented a record start flag which gives one additional depth but only as a Boolean value. So the resulting hierarchy model could be described by $b.x.x$ with $b \in \{true, false\}$ and $x \in \mathbb{N}$. The hierarchy analysis is much more based on the multi-topological grid than the ordering stage. This is because our diagonal ordering algorithm gives good initial results.

The hierarchy analysis stage consists of twelve basic rules and various combinations in sub routines. To deal with repeating record and indention structures the system uses two storage systems. One for the third hierarchy level and one for the first Boolean respectively second hierarchy level. For this we built a set of attribute classes based on the left position and the font size which is later looked up during the processing for determining the correct indention level. To identify possible main record candidates we defined a simple linear function as a hierarchy rule based on the font height as follows:

$$fontLimit = 11 + (0.25 \cdot maxFontHeight) \tag{5.12}$$

Example 5.2 shows a log output from the hierarchy detection stage introducing two additional rules. RULE10 refers to a fontHeight/fontWeight comparison and RULE 102 to an indention algorithm. The indention detection is not an easy task because humans use semantic deductions for interpreting indentions. For example: if one of the boxes is very small, probably no indention could be detected because there is no overlapping. We addressed this problem by doubling the width of very small boxes if the width ratio between the two boxes is below 0.1.

Example 5.2 Sample log output from the hierarchy process

```
*libHryDetClass* processHierarchy.process ...
  START - id:18
*libHryDetRSClass* rule ...
  RULE6 => YES ... aBox1.id:17 - aBox2.id:18;
    aBox1.fontHeight:20; aThreshold:20 (near)
  RULE10 => YES ... aBox1.id:17 - aBox2.id:18;
    aBox1.fontHeight:20; aBox2.fontHeight:19
    (font greater/equal)
  RULE102 => YES ... aBox1.id:17 - aBox2.id:18;
    width-ratio:0.22832369942196531 (indent ex)
  => YES
*libHryDetRSClass* rule ...
  rule1 => no ... aBox1.id:18 - aBox2.id:17
  (aligned)
  => no
```

5.9 Experiments

5.9.1 Data Selection and Ground Truthing

The experiments were applied to 85 web pages from four different domains: (1) search engine results, (2) personal homepages, (3) blog pages and as an addition with fewer pages (4) online newspapers. Some web pages were not considered because of overall complex layout hierarchy (probably even for humans) and lack of font size dependent structure. Two additional webpages could not be analyzed by the box identification algorithm. Due to the limitations of the REDEVILA system we removed also table or calendar structures by deactivating the corresponding boxes. Inside the

search domain we applied different search terms (“New York Times”, “BBC”, “Time Magazine”, “USA Today” and “Financial Times”) and a term for single records (see also the Googlewhack[7] web page) and used multiple web pages from the same search engine if the visual structure differs (e.g. search results with indention vs. without).

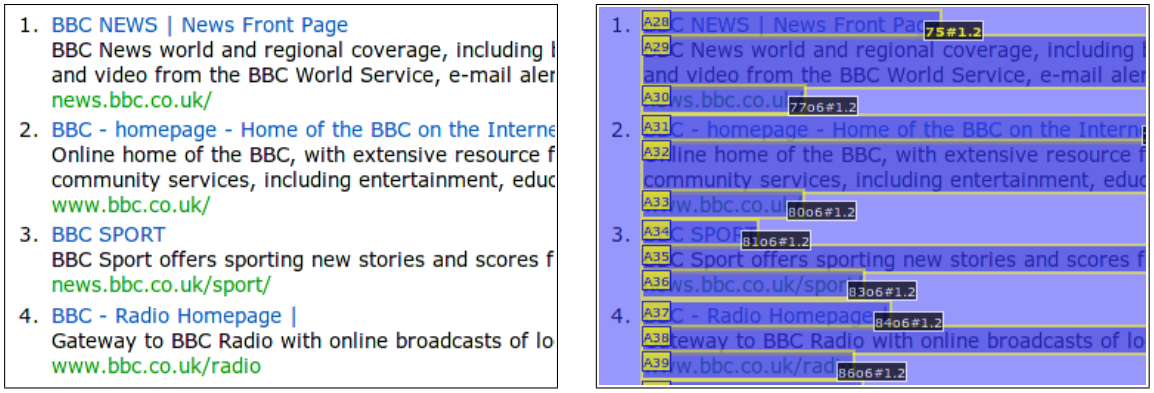


Figure 5.25: Example for failed record detection because of no distance and same font height[8]

Figure 5.25 shows an example of a removed web page. The fontsize of the headers and the explaining text is the same and there is no distance between the records which prevents the correct separation between the records. Another problem is the use of the `<ol>` tag for the list because the resulting numbers are not created as detectable seperate DOM nodes. An extension of the box detection algorithm by analyzing the parent DOM nodes would allow the correct separation. This is also an example where it would be reasonable to use tag information to improve the segmentation process.

Figure 5.26: Example for failed record detection because of same font height[9]

The webpage in figure 5.26 was excluded because of the same fontsize between headers and content. The only possible local based distinction would be the text color.

[7] http://www.googlewhack.com
[8] http://www.rectifi.org.uk/websearch2/BBC
[9] http://www.archpaper.com

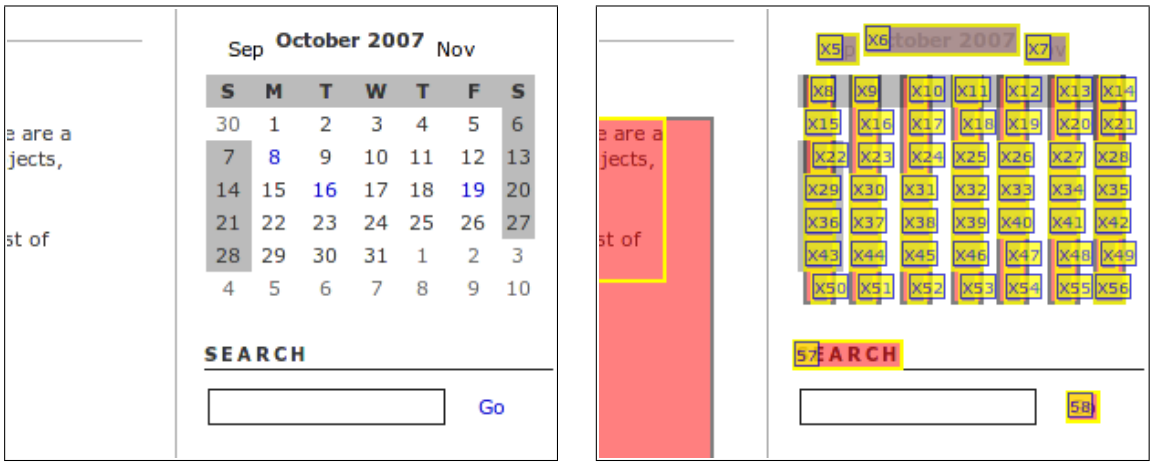


Figure 5.27: Example for box deactivation of a calendar area[10]

Figure 5.27 is an example for the deactivation of boxes of a calendar area. Calendars have a very specific layout and probably have to be described and analyzed by a seperate model. Gatterbauer et.al. [59] describes such structures as aligned substructured graphics (see also figure 1.3).

Our primary target was the record detection with the hierarchy as less important. Therefore we accepted small differences in the hierarchy analysis as long as the record was correctly detected. For example, if the hierarchy level after a first level header was set to 1.3 instead of 1.2 this was tolerated. The same holds true if the main header of a blog has the same hierarchy as the blog entry headers. Of course depending on the specific domain and web page there was a less or more ambiguity of what is a record and what not. Also the lack of an exact defined basic model made the decisions difficult. Therefore the experiments and the REDEVILA system with its setup present only one possible interpretation of the web page structure.

5.9.2 Automated Test Setup

For the automated test setup we used our VTXServer extension. This component provides a scriptable telnet interface for automating the whole process and was originally developed for the online VENTEX system. We used the extension for the mass generation of ARFF data files for the segment classification and applying the automatic approach at the ground truthed web pages.

```
baer@router_service:~/.Trash/figures_low$ telnet localhost 4444
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
>> ---------------------------------------
>> |    Welcome to the DBAI VTX Server    |
>> |        Version 0.2 - 07/05/08        |
>> ---------------------------------------
>> plugins: VTX_VACIE, VTX_VENTEX, VTX_WEBPAGEDUMP
>> availiable commands: LOAD, HELP, EXIT/QUIT, SET, SETDIR, SETFILE, ECHO, FOR/ROF,
EXEC, APPEND, SHUTDOWN/SD, VAC, VTX, WPD
>>

:$
```

Figure 5.28: The VTXServer telnet interface with the availiable commands

[10] http://www.sqljunkies.com/WebLog/marathonsqlguy

Figure 5.28 shows the telnet interface of the VTXServer extension and example 5.3 the basic script used for the WEKA ARFF file generation which was the input for the importance classification.

Example 5.3 VTXServer script for the ARFF file generation

```
set SEGDIR=/REDEVILA/LearningData/
vac arff A0 $SEGDIR$segment.arff
vac arff A1 $SEGDIR$boxes.arff
setdir SEGTEST=$SEGDIR$
for $SEGTEST$
  load $SEGTEST$index.html
  vac xmlload X1
  vac arffsave A0
  vac arffsave A1
  append $SEGTEST$vacie/arff_segment.txt $SEGDIR$segment.arff
  append $SEGTEST$vacie/arff_boxes.txt $SEGDIR$boxes.arff
rof
```

The server provides a simple `for` loop command and handles timeout issues during the loading of web pages automatically. Also a plugin interface is included for defining additional commands using the command line infrastructure of the server. For example the `vac` command is from a plugin for the REDEVILA extension and calls the ARFF and XML loading respectively saving routines. Because the loading and mass execution is handled by the server the plugin functions need only be designed for the single case.

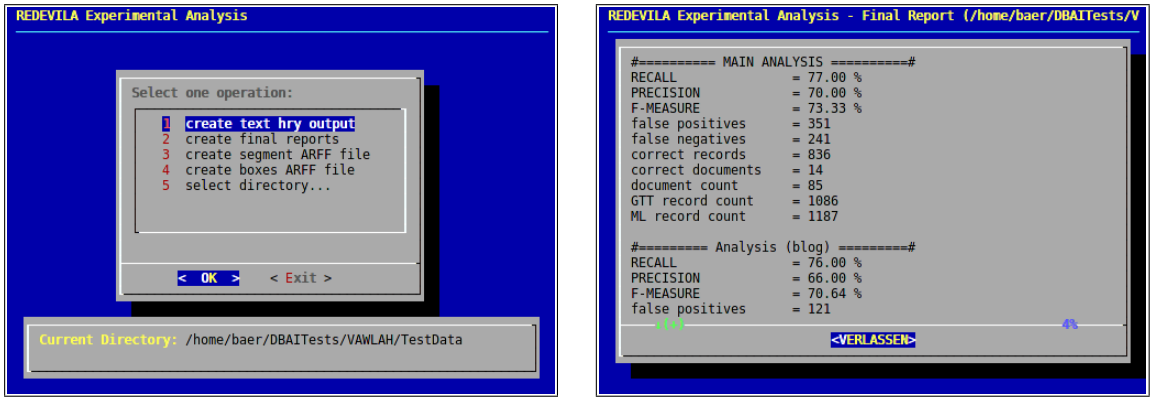


Figure 5.29: The REDEVILA analysis tool with the main menu and the automatic recall, precision and f-measure calculation

For fast experimental evaluation we set up a bash script with the basic `dialog` command providing a simple user interface. The determination of the false positives/negatives and correct records was based on the standard unix `diff` utility. Because we needed a line based comparison we used the `xsltproc` command line utility for the XSLT transformation of the XML hierarchy files into simple text files where each line corresponds to one single record. By analyzing the diff output it is easy to get the count of false positive, false negative and correct records.

A left insertion from the ML file (REDEVILA classification output) into the GTT file (ground truthing) counts as false positive and a right insertion from the GTT file into the ML file as a false negative (see figure 5.30 for the basic concept). The record count from both the ML and the GTT itself was read from the basic XML files. Together with the VTXServer extension it was possible to make a full automatic analysis after e.g. a rule change and see the effect of this specific change.

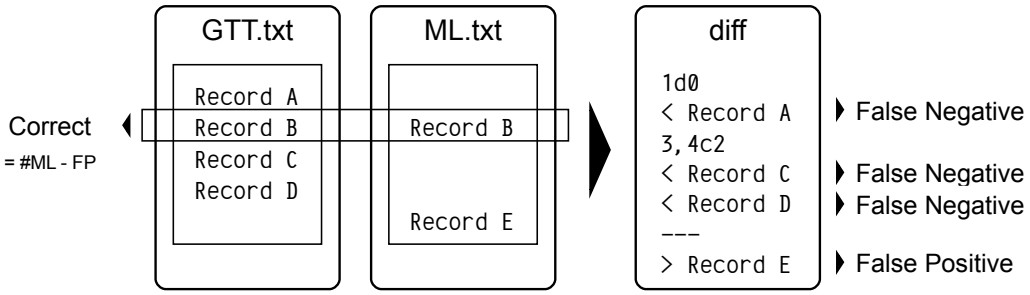


Figure 5.30: The automatic diff based evaluation concept

Table 5.6: Experimental results

Recall	Precision	F-Measure	Record Count	Correct Records	Document Count	Correct Documents
			General			
0.77	**0.70**	**0.73**	1086	836	85	14
			Blogs			
0.76	0.66	0.71	323	243	25	4
			Homepages			
0.79	0.77	0.78	231	184	25	10
			Search			
0.83	0.72	0.77	276	230	25	0
			Newspaper			
0.71	0.66	0.68	256	179	10	0

Figure 5.29 shows the analysis tool with the main menu on the left and a resulting analysis on the right. The automatic creation of the summary ARFF files (both boxes and segments) for the WEKA machine learning toolkit was also integrated.

5.9.3 Test Results

Table 5.6 shows the experimental results. Similar to our previous table extraction system the seemingly inferior result have to be interpreted with the overall lower precision of general visual based methods in mind. The *Homepages* domain has the best results which could be explained by the fact that many of this pages are designed in a very simple way with different font heights and easy interpretable distances. The *Newspaper* domain in contrast is much more difficult which is also expressed through the zero count of complete correctly identified web pages. Due to the wide variety of layout and design visual approaches could only be an addition to traditional methods as stated in the introduction. Generating specific domain dependent visual rules e.g. for newspapers would give better results. Of course there is a limitation regarding the visual complexity of web pages especially where sometimes even humans have difficulties in interpreting.

Semantic Word Level

Sometimes the semantic word content would be needed for a correct record detection. Consider, for example, the date entries of many blog entries which are located above the entry headers. Many blogs have only one entry per day or the underlying blog engine would generate a date for every entry independent of the count of messages per day. Because of this the REDEVILA system contains rules for detecting small text lines above headers and changes the order so the resulting order is switched and the small text line is ordered after the header line. Figure 5.31 shows a working example of the blog domain.

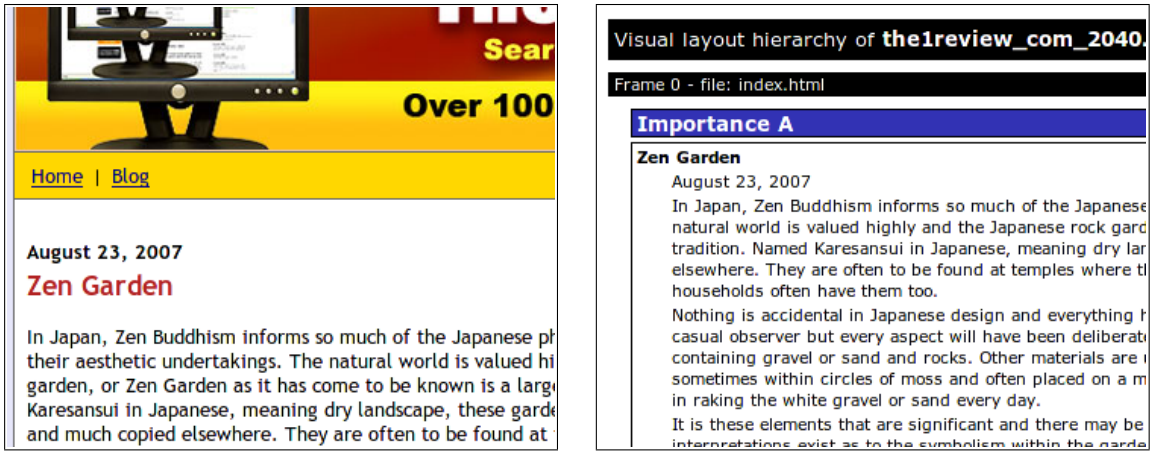


Figure 5.31: An example of the date rule for the blog domain with the original page on the left and the resulting hierarchy XML on the right[11]

Of course this is a kind of domain dependent rule because the position of the small text line above a header does not generally conclude a single relation. Figure 5.32 shows an example of a parent with multiple subhierarchy relations regarding the "Sponsor Results". The "small line above header" rule merges the "Sponsor Results" into the first record.

Another example of a semantic word level dependence is shown in figure 5.33. The picture description is related to the news message below but this could not be detected by a visual only approach.

[11] http://the1review.com

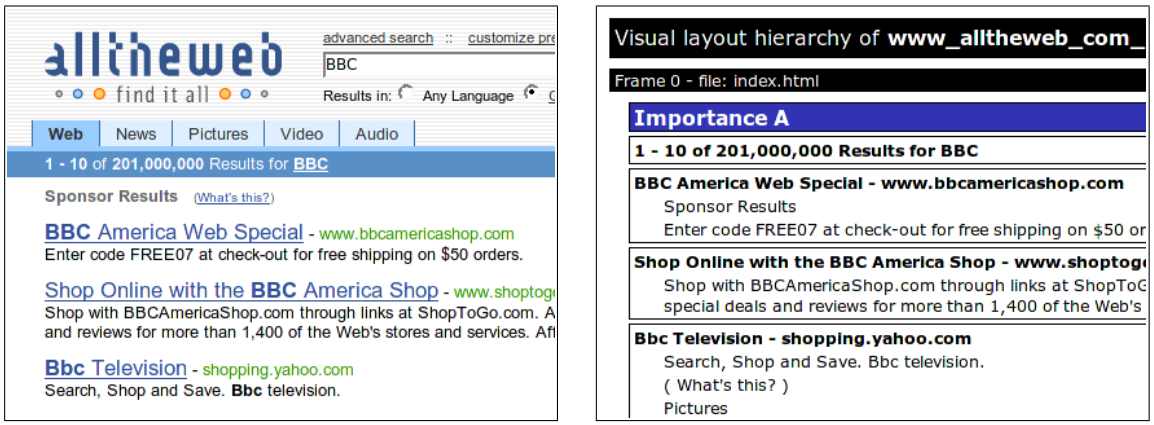


Figure 5.32: An example of the "small line above header" rule with the original page on the left and the resulting hierarchy XML on the right[12]

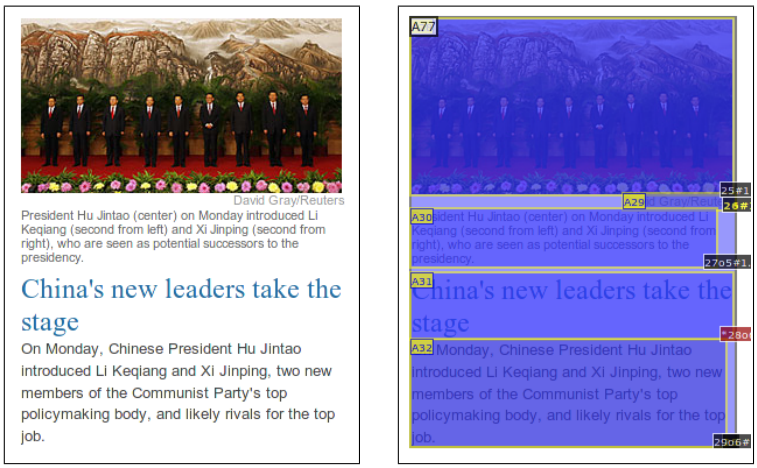


Figure 5.33: An example of a word semantic dependence with the original page on the left and the resulting segmentation on the right[13]

Single Records

The detection of single records is principally possible but has some limitations because of the sensitivity due to the small size of the web page which results in segment classification errors. Figure 5.34 shows an example of a single search record. The start of the record is correctly determined but the term "AltaVista found 1 results." is an example of word semantic or even domain dependence because it is added to the search record due to the "small line above" rule derived from the blog domain mentioned before. We could remove the rule which would give a perfect result but at the same time would reduce the performance for the blog domain. Nevertheless we counted this single search record as correct because the start and the end of the record is correctly determined and a processing of the result at word semantic level would be able to clean the result.

[12] http://www.alltheweb.com
[13] http://www.csmonitor.com

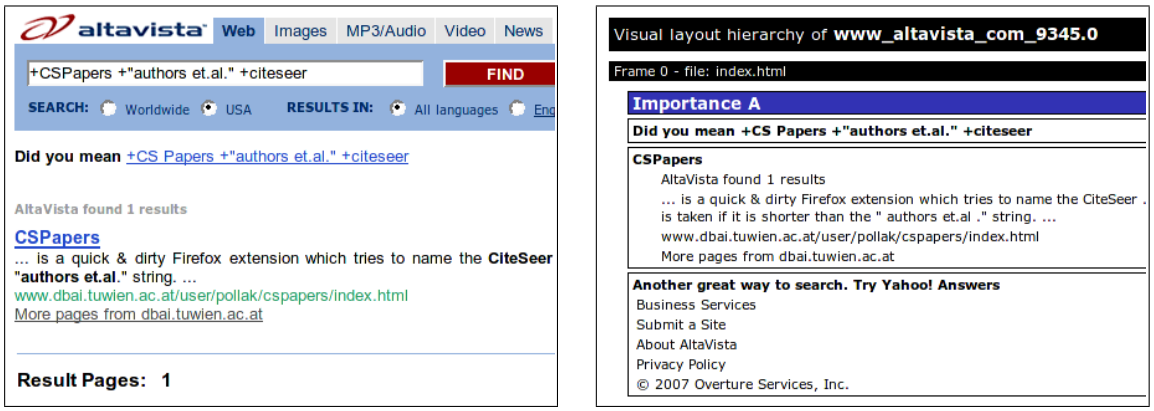


Figure 5.34: An example of a single record detection with the original page on the left and the resulting hierarchy XML on the right[14]

Another problem is the classification of noisy and important segments. Beside the ambiguity of what is important most people would agree that the two additional records ("Did you..." and "Another great...") are not that important and should be classified as noisy blocks. Again this could be achieved by a more domain dependent classification and/or the introduction of word level semantics. Probably also a different classification algorithm (e.g. Support Vector Machines instead of PARTS) would improve the results. Figure 5.35 shows a really bad result for a single record because of the small font size differences and some segmentation difficulties. But this problem holds true only for the first or the single record as the right side shows with multiple records from the same search engine.

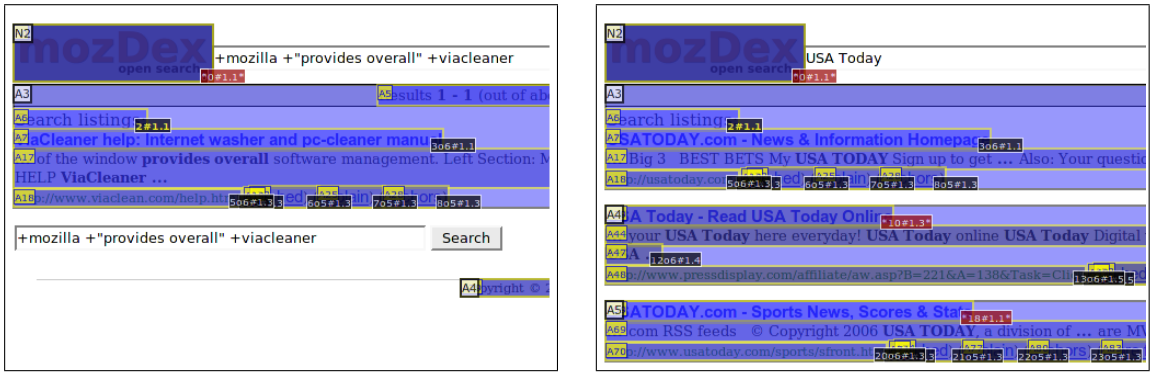


Figure 5.35: An example of a failed single record detection on the left and a multiple records result on the right[15]

A similar problem shows figure 5.36. If the "Did you mean..." line would be removed the record detection would be perfect. Nevertheless the results are valuable if processed further. For the search eninge domain the application of word level semantic rules would remove the "Did you mean..." and "Results 1 of 1..." entries. For such a visual analysis tool like the REDEVILA system it is important to have more false positives than false negatives because the false positives could be

[14]http://www.altavista.com
[15]http://www.mozdex.com

corrected, the false negatives are lost. The experimental results show about a quarter less false negatives than false positives.

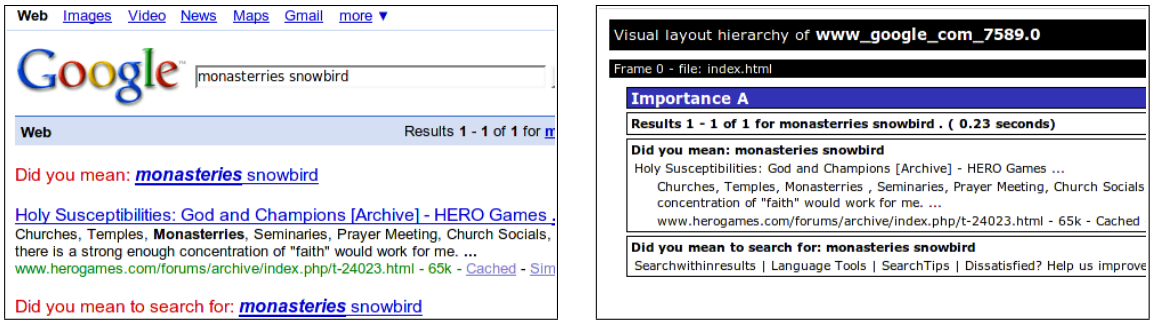


Figure 5.36: An example of a single record detection with the original page on the left and the resulting hierarchy XML on the right[16]

[16] http://www.google.com

Working Examples from Each Domain

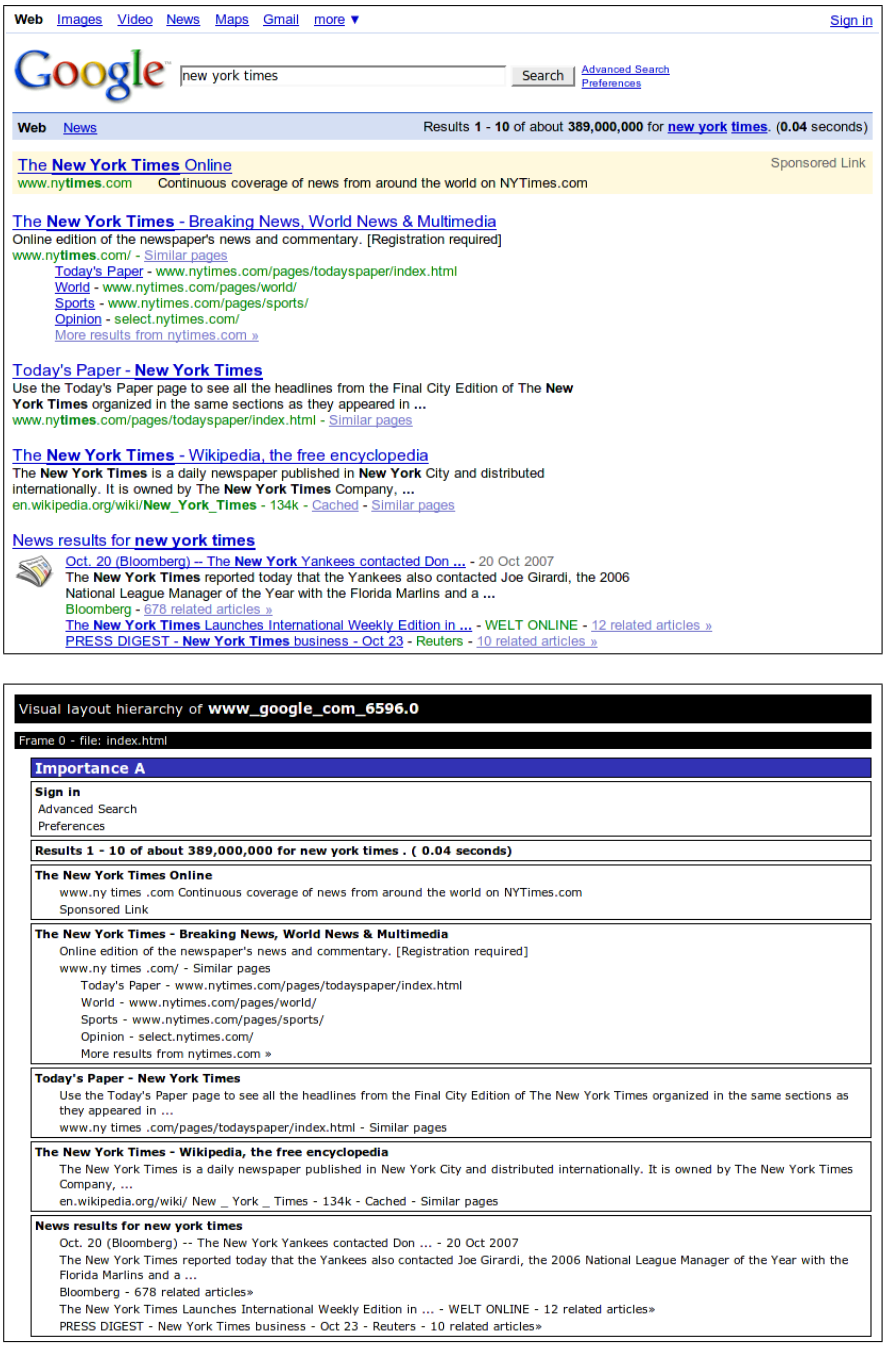


Figure 5.37: An example of a good hierarchy and record detection from the search domain with the original page on top and the resulting hierarchy XML below[17]

[17] http://www.google.com

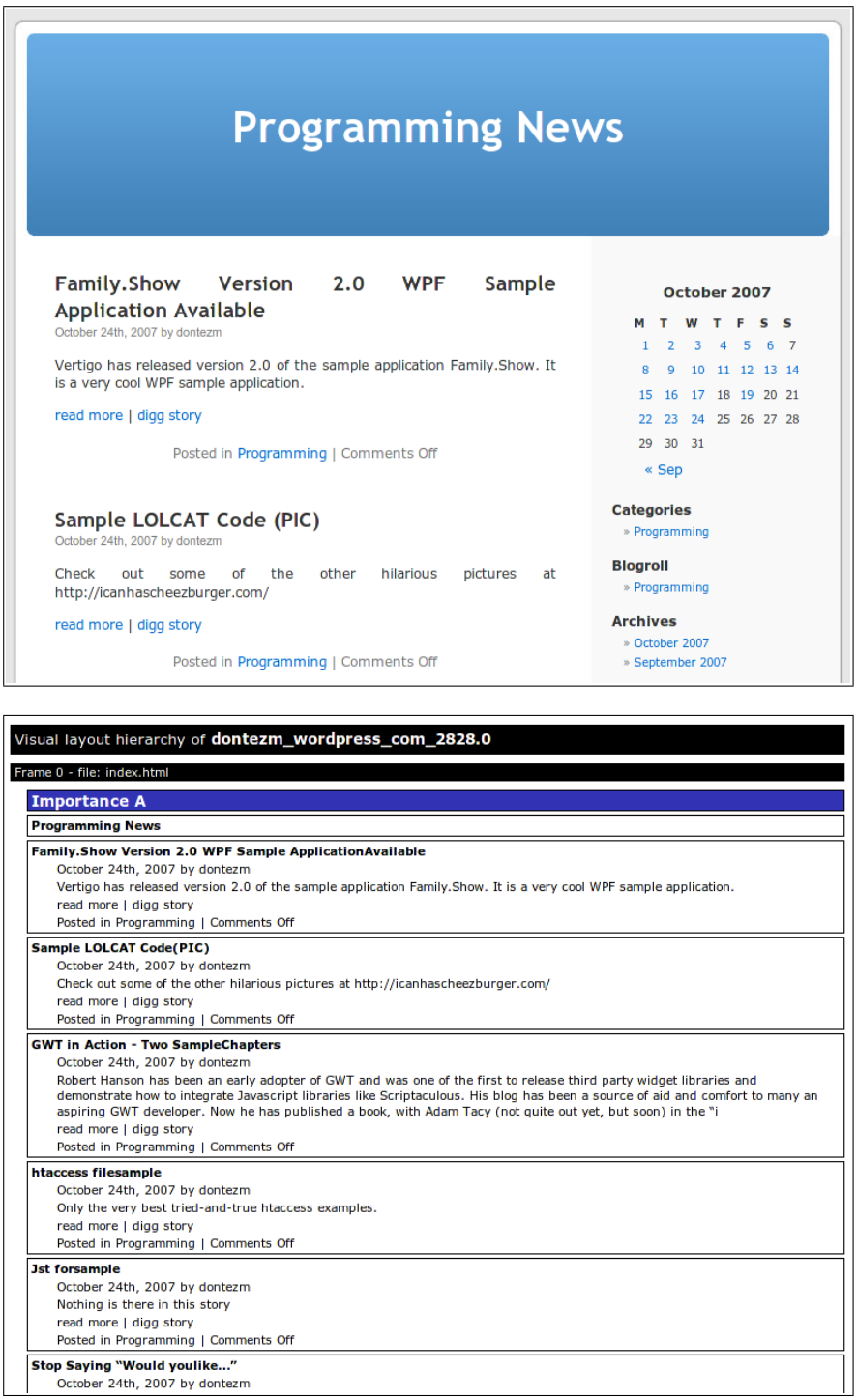


Figure 5.38: An example of a good hierarchy and record detection from the blog domain with the original page on top and the resulting hierarchy XML below[18]

[18]http://dontezm.wordpress.com

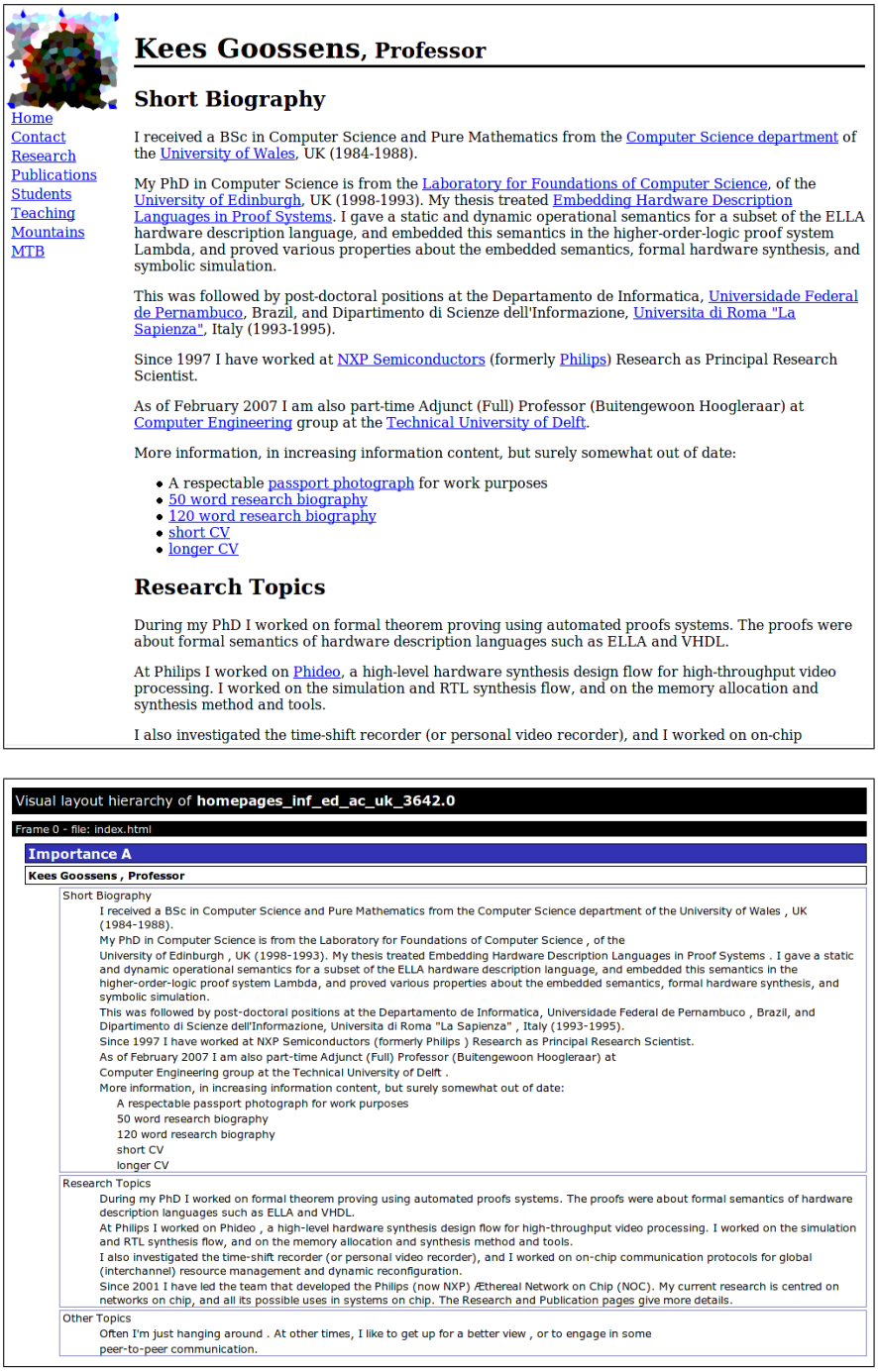


Figure 5.39: An example of a good hierarchy and record detection from the personal homepage domain with the original page on top and the resulting hierarchy XML below[19]

[19]http://homepages.inf.ed.ac.uk/kgoossen

Figure 5.40: An example of a good hierarchy and record detection from the newspaper domain with the original page on top and the resulting hierarchy XML below[20]

[20]http://www.independent.co.uk

6 Conclusions

This thesis was motivated by the observation that data records on web pages are structured not only by word content but also by an implied visual hierarchy. A model of this visual hierarchy can greatly help automatic information extraction approaches become more domain independent and robust against variations of HTML syntax changes because the analysis is applied only at the visual representation layer and that has to remain constant in a way understandable by humans.

In this thesis, we presented the REDEVILA (REcord DEtection on the VIsual LAyer) system which is capable of detecting records on web pages using visual analysis of web page layout hierarchies based on a visual functional semantic model. The system is principally domain independent as long as the layout hierarchy provided by the web page depends mainly on font size, distance and indention. We further proposed a diagonal ordering algorithm to obtain a more natural ordering and demonstrated the basic concept of the visual based detection of single records.

For the experimental evaluation we selected 85 web pages from four different domains (blogs, search results, personal homepages and newspapers) to show the basic domain independence of our system. Experiments were performed against manually annotated semantic hierarchies and achieved a fair overall performance (Recall: 0.77, Precision: 0.70, F-Measure: 0.73). When interpreting the results, the general lower performance of domain independent visual based approaches in contrast to traditional wrapper technologies, which are targeted and trained towards very specific domains, has to be taken into account (see also [14]). Beside this, the concept of redundancy of information would help to improve the accuracy of retrieved erroneous but multiple similar results during the information integration stage [57].

6.1 Discussion and Limits

Beside the principal domain independence of the visual approach there is a trade-off between domain independence and the correctness of the results as with all extraction methods. Nevertheless the level of domain independence is much higher than with traditional tag based systems. For example, the use of a specific rule for the blog domain in the experiments improved the blog domain results while lowering the search domain results although this rule is mainly applicable to the whole blog domain. If this generic visual concept is applied in a more domain dependent manner, (e.g. search records) the results would be usable inside a productive environment especially when combined with some word level semantic analysis. We could therefore redraw the graphic from the introduction as shown in figure 6.1 as this should be the future target for the visual web page analysis based on functional semantics.

The framework presented in this thesis is far from perfect because it was conceived as a first raw prototype system. Every stage of the analysis process could definitely be improved. The box merging algorithm operates on tag level which is a disadvantage because visually close but DOM tree distant elements are not merged.

The segmentation should be improved for detecting the basic visual layout grid which was used by the web designer and not the exact dimensions of the text boxes. Consider the case of different widths of left aligned records which would than be enclosed by segments with the same width

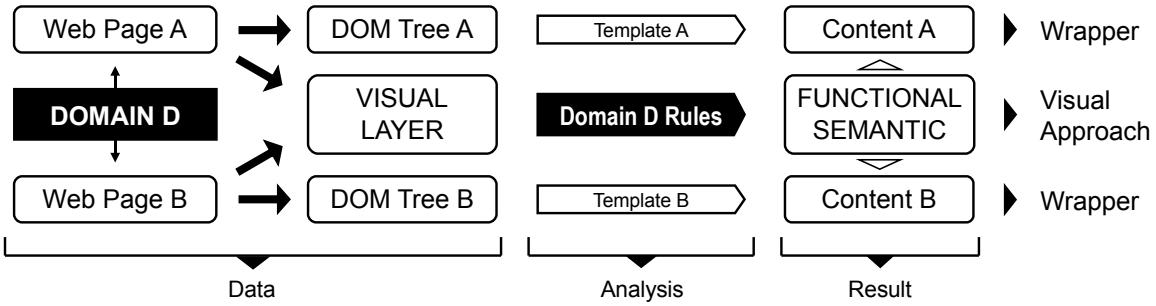


Figure 6.1: General difference between a traditional wrapper and a domain dependent visual approach

making the spatial reasoning much easier because simply the width has to be compared instead of a fuzzy left alignment check.

The improvement of the segmentation would reduce the importance for a global title font classification but at the same time this font classification could be improved by introducing either a machine learning approach or by providing a non-linear function together with a font size distribution analysis as titles and content text are normally distributed differently. It would also make sense to consider color issues during the analysis.

The multitopological grid forms a good base for the spatial reasoning process but the specific rules were generated by a human trial and error approach. The resulting rules give an insight in visual hierarchy structures but are surely not perfect. While the automatic test suite allows a relatively fast check if a specific rule lowers the correctness of the test web pages it would make sense to use also a machine learning approach similar to the importance classification to optimize the rules and to discover probably new visual rules and insights.

To make a point, we could subsume the above remarks by saying that the REDEVILA system together with the *visual rule based approach* is promising, but the focus should be shifted to domain dependent functional semantics for a productive application.

6.2 Future Work

Based on the previous discussion the approach presented in this work could be developed further by looking at the following issues:

Automatic rule generation: It would be interesting to use a machine learning approach for the spatial rules instead of the rule generation through humans. Various spatial box relationships could then be annotated and learned by a classification algorithm based on basic spatial expressions like “below and near”

Adaptive segmentation: One disadvantage of the used segmentation algorithm is the overall compromise which can result in to low or to high segmentation granularity. An extension by applying an adaptive segmentation algorithm which will consider various web page features dynamically would improve the results.

Colored record headers: To date the REDEVILA system does not considering colors for the record header candidate selection which would be a valuable extension. For this the feature distribution of the text blocks has to be analyzed further.

Complexity classification: a more objective classification criterion for the visual complexity of a web page has to be found to provide a better base for the data selection and the interpretation of the results.

Consideration of the visual layout grid: the consideration of the basic visual layout grid (e.g. search webpages with adds on the left, adds on the left and right or adds on the right only and also news pages with many columns vs. personal homepages with only one single column) would improve the block classification which does not distinguish between e.g. adds on the left and small left oriented content columns.

Systematic substructure model: similar to the table domain a more systematic model for substructered list has to be found to have a clear definition for the possibilities of a specific substructure related (visual) algorithm.

Integration of table models: the synthesis of the table model with the substructered list model into a general table/substructure model would allow to include both visual structures in the analysis process reducing the errors through tables for the substructured list processing and vice versa.

Use of tag information: after the *Absolute Positioning Safe* definition which allows a clear distinction between visual and tag based methods the use of tag information could take place in a more systematic manner and would improve the visual approach (see figure 5.25 for an example).

Distance relationship investigation: the ambiguity of distance descriptions is a serious problem and should be investigated further to get a better foundation for terms like “near”, “very near” or “far”. The font-size dependent approach taken by this thesis is far from optimal.

Introducing domain dependence: because of the existence of informal visual rules regarding different domains the focus should be shifted to domain dependent functional semantic (e.g. search records or blogs) to improve the results at the cost of less independence. Alternatively a library of dynamic domain dependent enhancers could be provided.

A WEKA Output

A.1 Final Feature Set

```
=== Run information ===

Scheme:       weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Relation:     segimphry-
  weka.filters.unsupervised.attribute.Remove-R1-
  weka.filters.supervised.attribute.AttributeSelection-E
  weka.attributeSelection.CfsSubsetEval-S
  weka.attributeSelection.BestFirst -D 1 -N 5
Instances:    370
Attributes:   6
              fontHeight
              leftPos
              topPos
              widthRatio
              charCount
              importance
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

PART decision list
------------------

widthRatio > 0.22 AND
leftPos <= 0.31 AND
fontHeight <= 0.59: A (66.0)

widthRatio <= 0.12 AND
leftPos > 0.61: N (55.0)

widthRatio > 0.3 AND
fontHeight > 0.61: A (53.0/2.0)

leftPos <= 0.61 AND
widthRatio <= 0.12 AND
fontHeight > 0.48 AND
charCount <= 81: N (48.0)

leftPos <= 0.61 AND
fontHeight > 0.4 AND
topPos <= 0.85 AND
widthRatio > 0.11 AND
widthRatio <= 0.21: A (38.0/2.0)

leftPos > 0.61: N (34.0)

fontHeight <= 0.4: N (19.0)

topPos > 0.83: N (13.0)

leftPos > 0.33 AND
fontHeight <= 0.54: A (9.0)

topPos <= 0.37: N (16.0)

charCount <= 87 AND
widthRatio > 0.08: N (10.0/1.0)

: A (9.0/1.0)

Number of Rules  :  12
```

```
Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         336               90.8108 %
Incorrectly Classified Instances        34                9.1892 %
Kappa statistic                          0.8155
Mean absolute error                      0.1108
Root mean squared error                  0.2844
Relative absolute error                 22.2901 %
Root relative squared error             57.048  %
Total Number of Instances              370

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.905     0.088      0.923     0.905     0.914    N
  0.912     0.095      0.891     0.912     0.902    A

=== Confusion Matrix ===

   a   b   <-- classified as
 180  19 |   a = N
  15 156 |   b = A
```

A.2 Initial Feature Set

```
=== Run information ===

Scheme:       weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Relation:     segimphry-
  weka.filters.unsupervised.attribute.Remove-R1-
  weka.filters.supervised.attribute.AttributeSelection-E
  weka.attributeSelection.CfsSubsetEval-S
  weka.attributeSelection.BestFirst -D 1 -N 5
Instances:    370
Attributes:   7
              leftPos
              topPos
              widthRatio
              charCount
              wordCount
              wordLinkCount
              importance
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

PART decision list
------------------

charCount <= 24 AND
topPos <= 0.8 AND
widthRatio <= 0.17 AND
topPos <= 0.14 AND
widthRatio > 0.01: N (56.0/1.0)

leftPos > 0.61 AND
wordCount <= 5 AND
charCount <= 7: N (26.0)

leftPos > 0.61 AND
wordLinkCount > 5: B (15.0)

widthRatio > 0.22 AND
leftPos <= 0.44 AND
topPos <= 0.93 AND
charCount > 79: A (93.0)

leftPos > 0.61 AND
widthRatio <= 0.18 AND
topPos <= 0.81 AND
leftPos > 0.63 AND
topPos > 0.19 AND
```

```
leftPos <= 0.83 AND
widthRatio <= 0.15: B (6.0)

wordLinkCount > 9 AND
widthRatio > 0.1: A (31.0/7.0)

topPos > 0.91 AND
widthRatio <= 0.42: N (20.0)

leftPos > 0.61 AND
widthRatio <= 0.16: N (8.0)

leftPos > 0.61 AND
leftPos <= 0.77: B (6.0)

widthRatio > 0.12 AND
leftPos <= 0.59 AND
wordCount > 6 AND
topPos > 0.02 AND
leftPos > 0.11: A (23.0/1.0)

wordLinkCount <= 9 AND
topPos > 0.52 AND
wordCount <= 6 AND
widthRatio > 0.18: A (9.0)

wordLinkCount <= 9 AND
topPos > 0.42 AND
wordCount <= 6 AND
topPos <= 0.57: N (11.0)

wordLinkCount <= 9 AND
topPos > 0.53 AND
wordLinkCount <= 4 AND
leftPos <= 0.37 AND
wordCount <= 2 AND
wordLinkCount > 1: N (6.0/1.0)

wordLinkCount <= 9 AND
topPos > 0.53 AND
wordCount <= 6 AND
leftPos > 0.09 AND
widthRatio > 0.02: A (6.0)

topPos > 0.61 AND
wordLinkCount <= 10: N (10.0/1.0)

topPos <= 0.02 AND
wordLinkCount <= 4 AND
topPos <= 0: A (2.0)

topPos <= 0.02 AND
topPos <= 0: N (2.0)

widthRatio <= 0.26 AND
wordCount > 9: B (10.0/1.0)

widthRatio > 0.19: A (6.0)

topPos <= 0.25 AND
topPos > 0.02 AND
wordCount <= 1: B (5.0)

topPos <= 0.25 AND
wordLinkCount > 3: B (4.0/1.0)

leftPos <= 0.02 AND
topPos <= 0.45: B (4.0)

topPos > 0.25: A (8.0/1.0)

: N (3.0)

Number of Rules  :  24


Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
```

```
=== Summary ===

Correctly Classified Instances         309               83.5135 %
Incorrectly Classified Instances        61               16.4865 %
Kappa statistic                          0.7301
Mean absolute error                      0.1313
Root mean squared error                  0.3151
Relative absolute error                 32.0387 %
Root relative squared error             69.6378 %
Total Number of Instances              370

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.846     0.119      0.818     0.846     0.832    N
  0.889     0.095      0.889     0.889     0.889    A
  0.643     0.048      0.706     0.643     0.673    B

=== Confusion Matrix ===

   a   b   c   <-- classified as
 121  12  10 |   a = N
  14 152   5 |   b = A
  13   7  36 |   c = B
```

A.3 ZeroR Classifier

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     segimphry-
  weka.filters.unsupervised.attribute.Remove-R1-
  weka.filters.supervised.attribute.AttributeSelection-E
  weka.attributeSelection.CfsSubsetEval-S
  weka.attributeSelection.BestFirst -D 1 -N 5
Instances:    370
Attributes:   6
              fontHeight
              leftPos
              topPos
              widthRatio
              charCount
              importance
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

ZeroR predicts class value: N

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         199               53.7838 %
Incorrectly Classified Instances       171               46.2162 %
Kappa statistic                          0
Mean absolute error                      0.4972
Root mean squared error                  0.4986
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              370

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1         1          0.538     1         0.699    N
  0         0          0         0         0        A

=== Confusion Matrix ===

   a   b   <-- classified as
 199   0 |   a = N
 171   0 |   b = A
```

B Test Output

```
#========== MAIN ANALYSIS ==========#
RECALL               = 77.00 %
PRECISION            = 70.00 %
F-MEASURE            = 73.33 %
false positives      = 351
false negatives      = 241
correct records      = 836
correct documents    = 14
document count       = 85
GTT record count     = 1086
ML record count      = 1187

#========= Analysis (blog) =========#
RECALL               = 76.00 %
PRECISION            = 66.00 %
F-MEASURE            = 70.64 %
false positives      = 121
false negatives      = 76
correct records      = 243
correct documents    = 4
document count       = 25
GTT record count     = 323
ML record count      = 364

#======= Analysis (homepage) =======#
RECALL               = 79.00 %
PRECISION            = 77.00 %
F-MEASURE            = 77.98 %
false positives      = 52
false negatives      = 47
correct records      = 184
correct documents    = 10
document count       = 25
GTT record count     = 231
ML record count      = 236

#======== Analysis (search) ========#
RECALL               = 83.00 %
PRECISION            = 72.00 %
F-MEASURE            = 77.10 %
false positives      = 89
false negatives      = 46
correct records      = 230
correct documents    = 0
document count       = 25
GTT record count     = 276
ML record count      = 319

#====== Analysis (newspaper) =======#
RECALL               = 71.00 %
PRECISION            = 66.00 %
F-MEASURE            = 68.40 %
false positives      = 89
false negatives      = 72
correct records      = 179
correct documents    = 0
document count       = 10
GTT record count     = 256
ML record count      = 268

#============= File List ===========#
DIRECTORY,CAT,TESTID,SEARCH,GTT_RECA,ML_RECA,FPOS,FNEG,CORRECT
beat_bodoglife_com_2409.0,blog,GoBlSe, Computer Science,11,11,0,0,11
blog_lowesoftware_com_2792.0,blog,GoBlSe, Computer Science,11,8,3,6,5
digiplay_info_1972.0,blog,GoBlSe, Computer Science,11,14,6,3,8
disgodkidd_blogspot_com_2978.0,blog,GoBlSe, Computer Science,4,4,0,0,4
distributedneuron_net_3337.0,blog,GoBlSe, Computer Science,12,16,5,1,11
```

```
dontezm_wordpress_com_2828.0,blog,GoBlSe, Computer Science,11,11,1,1,10
dwarren14_blogspot_com_2788.0,blog,GoBlSe, Computer Science,8,11,6,3,5
freestudiesabroad_blogspot_com_3736.0,blog,GoBlSe, Computer Science,31,33,14,12,19
gilpin_wordpress_com_2702.0,blog,GoBlSe, Computer Science,10,15,7,2,8
homemadedegrees_blogspot_com_3499.0,blog,GoBlSe, Computer Science,9,7,2,4,5
jjinux_blogspot_com_2596.0,blog,GoBlSe, Computer Science,8,15,10,3,5
jordannunes_blogspot_com_3123.0,blog,GoBlSe, Computer Science,7,7,2,2,5
lambda-the-ultimate_org_2910.0,blog,GoBlSe, Computer Science,11,17,8,2,9
laptopbudget_wordpress_com_3350.0,blog,GoBlSe, Computer Science,17,24,8,1,16
lispy_wordpress_com_2620.0,blog,GoBlSe, Computer Science,11,11,0,0,11
niniane_blogspot_com_2670.0,blog,GoBlSe, Computer Science,7,5,4,6,1
techlun_ch_1651.0,blog,GoBlSe, Computer Science,11,11,0,0,11
telcom2935_blogspot_com_2787.0,blog,GoBlSe, Computer Science,14,17,9,2,8
thelreview_com_2040.0,blog,GoBlSe, Computer Science,8,5,4,7,1
userslib_com_1885.0,blog,GoBlSe, Computer Science,18,16,1,3,15
vmblog_com_2574.0,blog,GoBlSe, Computer Science,41,44,8,5,36
www_packetslave_com_2586.0,blog,GoBlSe, Computer Science,11,18,9,2,9
www_scottstonehouse_ca_3417.0,blog,GoBlSe, Computer Science,7,4,3,6,1
www_sqljunkies_com_5948.0,blog,GoBlSe, Computer Science,19,26,10,3,16
www_terminally-incoherent_com_4087.0,blog,GoBlSe, Computer Science,15,14,1,2,13

altman_casimirinstitute_net_3448.0,homepage,GoPeHo,personal homepage,1,1,0,0,1
astro_imperial_ac_uk_3290.0,homepage,GoPeHo,personal homepage,7,7,4,4,3
dsrg_mff_cuni_cz_2900.0,homepage,GoPeHo,personal homepage,6,6,0,0,6
feynman_mit_edu_4401.0,homepage,GoPeHo,personal homepage,6,5,1,2,4
gnuhh_org_3076.0,homepage,GoPeHo,personal homepage,9,17,11,3,6
homepages_inf_ed_ac_uk_3642.0,homepage,GoPeHo,personal homepage,4,4,0,0,4
matteocorti_ch_2091.0,homepage,GoPeHo,personal homepage,6,7,1,0,6
people_brandeis_edu_3160.0,homepage,GoPeHo,personal homepage,16,23,12,5,11
www_acoustics_hut_fi_3259.0,homepage,GoPeHo,personal homepage,5,5,0,0,5
www_astro_su_se_2965.0,homepage,GoPeHo,personal homepage,1,1,0,0,1
www_balasko_com_2148.0,homepage,GoPeHo,personal homepage,16,14,1,3,13
www_ccnl_emory_edu_2900.0,homepage,GoPeHo,personal homepage,10,9,2,3,7
www_davelane_ca_2124.0,homepage,GoPeHo,personal homepage,4,5,2,1,3
www_dcs_qmul_ac_uk_3191.0,homepage,GoPeHo,personal homepage,7,8,2,1,6
www_fang_ece_ufl_edu_2546.0,homepage,GoPeHo,personal homepage,10,10,1,1,9
www_fiftythree_org_3523.0,homepage,GoPeHo,personal homepage,3,3,0,0,3
www_math_psu_edu_3027.0,homepage,GoPeHo,personal homepage,7,7,0,0,7
www_mrl_nott_ac_uk_3097.0,homepage,GoPeHo,personal homepage,3,3,0,0,3
www_nuff_ox_ac_uk_3576.0,homepage,GoPeHo,personal homepage,40,33,5,12,28
www_srcf_ucam_org_2934.0,homepage,GoPeHo,personal homepage,6,6,1,1,5
www_thomaskho_com_2873.0,homepage,GoPeHo,personal homepage,5,4,2,3,2
www_wu_ece_ufl_edu_2370.0,homepage,GoPeHo,personal homepage,22,22,1,1,21
www-leibniz_imag_fr_5421.0,homepage,GoPeHo,personal homepage,5,5,0,0,5
www-personal_umich_edu_3774.0,homepage,GoPeHo,personal homepage,9,9,0,0,9
www2_gsb_columbia_edu_4520.0,homepage,GoPeHo,personal homepage,23,22,6,7,16

en_epochtimes_com_3458.0,newspaper,GoNeSe,newspaper,13,11,2,4,9
www_ajc_com_1717.0,newspaper,GoNeSe,newspaper,32,44,19,6,25
www_azcentral_com_2379.0,newspaper,GoNeSe,newspaper,31,31,5,5,26
www_baltimoresun_com_2716.0,newspaper,GoNeSe,newspaper,36,35,10,11,25
www_chicagotribune_com_2894.0,newspaper,GoNeSe,newspaper,36,36,13,14,23
www_churchnewspaper_com_3936.0,newspaper,GoNeSe,newspaper,9,13,9,0,4
www_csmonitor_com_2405.0,newspaper,GoNeSe,newspaper,27,20,11,18,9
www_express_co_uk_2779.0,newspaper,GoNeSe,newspaper,16,21,7,2,14
www_independent_co_uk_2742.0,newspaper,GoNeSe,newspaper,20,20,4,4,16
www_washingtonpost_com_2959.0,newspaper,GoNeSe,newspaper,36,37,9,8,28

search_live_com_7054.0,search,SeEnRe,New York Times,14,14,2,2,12
search_yahoo_com_7961.0,search,SeEnRe,USA Today,11,12,5,4,7
search_yahoo_com_8369.0,search,SeEnRe,Time Magazine,25,32,8,1,24
websearch_cs_com_0614.0,search,SeEnRe,single record,1,1,1,1,0
websearch_cs_com_8198.0,search,SeEnRe,New York Times,20,23,3,0,20
websearch_cs_com_9239.0,search,SeEnRe,Financial Times,23,26,3,0,23
www_alltheweb_com_7774.0,search,SeEnRe,BBC,16,15,6,7,9
www_altavista_com_5772.0,search,SeEnRe,BBC,16,19,7,4,12
www_altavista_com_6362.0,search,SeEnRe,USA Today,23,25,7,5,18
www_altavista_com_9345.0,search,SeEnRe,single record,1,3,2,0,1
www_ask_com_5329.0,search,SeEnRe,BBC,11,16,7,2,9
www_exalead_com_6344.0,search,SeEnRe,single record,1,2,2,1,0
www_exalead_com_6918.0,search,SeEnRe,Time Magazine,11,12,2,1,10
www_exalead_com_7099.0,search,SeEnRe,New York Times,11,12,2,1,10
www_gigablast_com_4410.0,search,SeEnRe,BBC,10,10,1,1,9
www_gigablast_com_5511.0,search,SeEnRe,Time Magazine,10,11,2,1,9
www_gigablast_com_7162.0,search,SeEnRe,single record,3,2,2,3,0
www_goodsearch_com_4737.0,search,SeEnRe,BBC,18,22,4,0,18
www_goodsearch_com_7033.0,search,SeEnRe,single record,1,2,2,1,0
www_google_com_5322.0,search,SeEnRe,USA Today,10,14,4,0,10
www_google_com_5991.0,search,SeEnRe,Financial Times,13,17,5,1,12
```

```
www_google_com_6596.0,search,SeEnRe,BBC,11,15,5,1,10
www_google_com_7589.0,search,SeEnRe,single record,2,3,3,2,0
www_mozdex_com_5878.0,search,SeEnRe,USA Today,12,10,3,5,7
www_mozdex_com_9394.0,search,SeEnRe,single record,2,1,1,2,0
```

List of Figures

List of Tables

Bibliography

[1] O. T. Akindele and A. Belaïd. Page segmentation by segment tracing. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR)*, pages 341–344, October 1993. (Cited on pages 8 and 24.)

[2] Teruo Akiyama and Isao Masuda. A method of document-image segmentation based on projection profiles, stroke densities and circumscribed rectangles. In *Trans. of IEICE*, volume J69-D, pages 1187–1195, August 1986. (Cited on pages 7, 20 and 82.)

[3] Teruo Akiyama and Isao Masuda. A method of document-image segmentation based on projection profiles, stroke densities and circumscribed rectangles. *Systems and Computers in Japan*, 18(4):101–111, 1987. (Cited on page 7.)

[4] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983. (Cited on pages 29 and 82.)

[5] Oronzo Altamura, Floriana Esposito, and Donato Malerba. Wisdom++: An interactive and adaptive document analysis system. In *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 366, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on page 9.)

[6] A. Antonacopoulos and R.T. Ritchings. Flexible page segmentation using the background. In *Proceedings of the 12th IAPR International. Conference on Pattern Recognition - Conference B: Computer Vision and Image Processing.*,, volume 2, pages 339–344, 1994. (Cited on page 8.)

[7] Henrik Arndt. *Integrierte Informationsarchitektur*. Springer, Berlin, Heidelberg, 2006. (Cited on page 58.)

[8] Antoine Sourou Azokly. *Une approche uniforme pour la reconnaissance de la structure physique de documents composites fonde sur l'analyse des espaces*. PhD thesis, Universit de Fribourg (Suisse), 1995. (Cited on page 6.)

[9] Henry S. Baird and David J. Ittner. Language-free layout analysis. In *Procceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR), IAPR*, pages 336–340. IAPR, IEEE, October 1993. (Cited on page 12.)

[10] Henry S. Baird, S. E. Jones, and S. J. Fortune. Image segmentation by shape-directed covers. In *International Conference on Pattern Recognition*, 1990. (Cited on pages 7, 24 and 82.)

[11] Shumeet Baluja. Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 33–42, Edinburgh, Scotland, 2006. ACM Press. (Cited on pages 16, 26, 27 and 82.)

[12] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, Hyderabad, India, January 2007. (Cited on page 15.)

[13] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with lixto. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 119–128, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. (Cited on page 14.)

[14] Robert Baumgartner, Wolfgang Gatterbauer, and Georg Gottlob. Web data extraction systems. Encyclopedia of Database Systems. Springer, 2008. (Cited on page 72.)

[15] T. A. Bayer. Understanding structured text documents by a model-based document analysis system. In *Proc. of the 2nd ICDAR*, pages 448–453, October 1993. (Cited on page 11.)

[16] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, and Bernd Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2):74–82, 1992. (Cited on page 13.)

[17] Dan S. Bloomberg. Image analysis using threshold reduction. In *Proc. of SPIE, Image Algebra and Morphological Image Processing II*, pages 38–52, July 1991. (Cited on page 12.)

[18] Dan S. Bloomberg. Multiresolution morphological approach to document image analysis. In *Proc. of ICDAR*, pages 963–971, September/October 1991. (Cited on page 12.)

[19] Bert Bos, Håkon Wium Lie, Chris Lilley, and Ian Jacobs. Cascading style sheets, level 2 css2 specification. W3C Recommendation REC-CSS2-19980512, World Wide Web Consortium, May 1998. See http://www.w3.org/TR/REC-CSS2. (Cited on page 54.)

[20] Thomas M. Breuel. Layout analysis by exploring the space of segmentation parameters. In *Proc. of the 4th IAPR Workshop on Document Analysis Systems*, December 2000. (Cited on page 9.)

[21] Thomas M. Breuel. Two geometric algorithms for layout analysis. In *DAS '02: Proceedings of the 5th International Workshop on Document Analysis Systems V*, pages 188–199, London, UK, 2002. Springer-Verlag. (Cited on page 25.)

[22] Thomas M. Breuel. An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 66, Washington, DC, USA, August 2003. IEEE Computer Society. (Cited on page 25.)

[23] Thomas M. Breuel. High performance document layout analysis. In *Symposium on Document Image Understanding Technology*, 2003. (Cited on pages 30 and 82.)

[24] Thomas M. Breuel. Layout analysis based on text line segment hypotheses. In *3rd Int. Workshop on Document Layout Interpretation and its Applications (DLIA2003)*, August 2003. (Cited on pages 25, 30 and 82.)

[25] Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK, 1999. Springer-Verlag. (Cited on page 13.)

[26] Radek Burget. Hierarchies in html documents: Linking text to concepts. In *DEXA '04: Proceedings of the Database and Expert Systems Applications, 15th International Workshop on (DEXA'04)*, pages 186–190, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on page 15.)

[27] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Proc. 5th Asian-Pacific Web Conference (Web Technologies and Applications)*, pages 406–417. Springer, April 2003. (Cited on pages 3, 15, 16, 26 and 45.)

[28] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Vips: a vision-based page segmentation algorithm. Technical Report MSR-TR-2003-79, Microsoft Technical Report, November 2003. (Cited on pages 3, 15, 16, 26, 45 and 82.)

[29] Huaigu Cao, Rohit Prasad, Prem Natarajan, and Ehry MacRostie. Robust page segmentation based on smearing and error correction unifying top-down and bottom-up approaches. In *Proc. ICDAR*, September 2007. (Cited on page 9.)

[30] R. Cattoni, T. Coianiz, S. Messelodi, and C.M. Modena:. Geometric layout analysis techniques for document image understanding: a review. 9703-09, ITC-IRST, Trento, Italy, 1998. (Cited on page 6.)

[31] Cesarini, Lastri, Marinai, and Soda. Encoding of modified x-y trees for document classification. In *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 1131, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 21.)

[32] F. Cesarini, S. Marinai, G. Soda, and M. Gori. Structured document segmentation and representation by the modified x-y tree. In *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 563, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on page 21.)

[33] Chia-Hui Chang, Chun-Nan Hsu, and Shao-Cheng Lui. Automatic information extraction from semi-structured web pages by pattern discovery. *Decis. Support Syst.*, 35(1):129–147, 2003. (Cited on page 15.)

[34] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, and Khaled Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, October 2006. (Cited on page 6.)

[35] Sudarshan S. Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey D. Ullman, and Jennifer Widom. The tsimmis project: Integration of heterogeneous information sources. In *IPSJ*, pages 7–18, 1994. (Cited on page 13.)

[36] Kevin Chen, Stefan Jaeger, Guangyu Zhu, and David Doermann. Doclib - a document processing research tool. In *Proc. of SDIUT, Document Recognition and Retrieval XIII*, November 2005. (Cited on page 13.)

[37] H. Cheng, C. Bouman, and J. Allebach. Multiscale document segmentation. In *IS&T 50th Annual Conference*, pages 417–425, May 1997. (Cited on page 13.)

[38] A. G. Cohn and S. M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundam. Inf.*, 46(1-2):1–29, 2001. (Cited on pages 29 and 82.)

[39] Anthony G. Cohn. Qualitative spatial representation and reasoning techniques. In *Proc. of 21st Annual German Conference on Artificial Intelligence*, pages 1–30, London, UK, 1997. Springer-Verlag. (Cited on page 29.)

[40] Alan Conway. Page grammars and page parsing. a syntactic approach to documentlayout recognition. In *Proc. of the 2nd ICDAR*, pages 761–764, October 1993. (Cited on page 11.)

[41] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 109–118, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. (Cited on page 14.)

[42] Andreas Dengel and Gerhard Barth. Anastasil: A hybrid knowledge-based system for document layout analysis. In *IJCAI*, pages 1249–1254, 1989. (Cited on page 10.)

[43] Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Fein, and Frank Hones. From paper to office document standard representation. *Computer*, 25(7):63–67, July 1992. (Cited on pages 7 and 17.)

[44] Melvil Dewey. *A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library*. Amherst, Mass., 1876. from Project Gutenberg, http://www.gutenberg.org/etext/12513. (Cited on page 58.)

[45] David S. Doermann, Azriel Rosenfeld, and Ehud Rivlin. The function of documents. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 1077–1081, Washington, DC, USA, 1997. IEEE Computer Society. (Cited on pages 3, 5, 39 and 83.)

[46] Robert B. Doorenbos, Oren Etzioni, and Daniel S. Weld. A scalable comparison-shopping agent for the world-wide web. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 39–48, New York, NY, USA, 1997. ACM Press. (Cited on page 13.)

[47] Max J. Egenhofer and Robert Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991. (Cited on page 54.)

[48] Angela Engels. *Aufmerksamkeitsbasierte Lokalisierung und Bewertung relevanter Information auf Papierdokumenten*. PhD thesis, Fakultt fr Elektrotechnik und Informationstechnik , Technische Universitt Mnchen, 2000. (Cited on page 39.)

[49] F. Esposito, D. Malerba, and G. Semeraro. A knowledge-based approach to the layout analysis. In *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, page 466, Washington, DC, USA, 1995. IEEE Computer Society. (Cited on page 11.)

[50] Floriana Esposito, Donato Malerba, Giovanni Semeraro, Enrico Annese, and Giovanni Scafuro. An experimental page layout recognition system for office document automatic classification: An integrated approach for inductive generalization. In *In Proc. of the 10th ICPR*, volume 1, pages 557–562., June 1990. (Cited on pages 10 and 11.)

[51] K. Etemad, D. Doermann, and R. Chellappa. Document page decomposition by integration of distributed soft decisions. In *IEEE International Conference on Neural Networks*, volume 6, pages 4022–4027, June/July 1994. (Cited on page 12.)

[52] Kamran Etemad, David Doermann, and Rama Chellappa. Multiscale segmentation of unstructured document pages using soft decision integration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1):92–96, January 1997. (Cited on page 13.)

[53] Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proc. 19th AAAI / 16th IAAI*, pages 391–398, San Jose, California, July 2004. AAAI Press/The MIT Press. (Cited on page 15.)

[54] James L. Fisher, Stuart C. Hinds, and Donald P. D'Amato. A rule-based system for document image segmentation. In *Proc. of the 10th ICPR*, pages 567–572, June 1990. (Cited on pages 10, 17 and 18.)

[55] Andrew U. Frank. Formal models for cognition - taxonomy of spatial location description and frames of reference. In Christian Freksa, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, volume 1404 of *Lecture Notes in Computer Science*, pages 293–312. Springer, 1998. (Cited on pages 54, 55 and 83.)

[56] Gatos and Papamarkos. Applying fast segmentation techniques at a binary image represented by a set of non-overlapping blocks. In *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 1147, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 19.)

[57] Wolfgang Gatterbauer. *Contributions to large-scale information acquisition from the Web*. Ph.D. thesis, Vienna University of Technology, 2007. (Cited on pages 3 and 72.)

[58] Wolfgang Gatterbauer and Paul Bohunsky. Table extraction using spatial reasoning on the css2 visual box model. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, pages 1313–1318, Boston, MA, USA, July 2006. AAAI, MIT Press. (Cited on page 3.)

[59] Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. Towards domain-independent information extraction from web tables. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 71–80, Banff, Alberta, Canada, 2007. ACM Press. (Cited on pages 3, 4, 28, 43, 61 and 82.)

[60] Jaekyu Ha, R. M. Haralick, and I. T. Phillips. Recursive x-y cut using bounding boxes of connected components. In *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2)*, page 952, Washington, DC, USA, August 1995. IEEE Computer Society. (Cited on page 8.)

[61] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the web. In *Proceedings of the Workshop on the Management of Semistructured Data (in conjunction with ACM SIGMOD 1997)*, Tucson, Arizona, May 1997. (Cited on page 13.)

[62] Xiaolong Hao, Jason T. L. Wang, and Peter A. Ng. Nested segmentation: An approach for layout analysis in document classification. In *Proc. of the 2nd ICDAR*, pages 319–322, October 1993. (Cited on pages 8, 21, 27 and 82.)

[63] Robert M. Haralick. Document image understanding: Geometric and logical layout. In *CVPR94: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 385–390, 1994. (Cited on page 6.)

[64] Tamir Hassan and Robert Baumgartner. Using graph matching techniques to wrap data from pdf documents. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th international conference on World Wide Web, WWW 2006,*, pages 901–902. ACM, May 2006. (Cited on pages 27 and 82.)

[65] J. Higashino, H. Fujisawa, Y. Nakano, and M. Ejiri. A knowledge-based segmentation method for document understanding. In *ICPR86*, pages 745–748, 1986. (Cited on page 10.)

[66] Yuki Hirayama. A block segmentation method for document images with complicated column structures. In *Proc. of the 2nd ICDAR*, pages 91–94, October 1993. (Cited on pages 7 and 8.)

[67] Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Inf. Syst.*, 23(9):521–538, 1998. (Cited on page 14.)

[68] Yunhua Hu, Guomao Xin, Ruihua Song, Guoping Hu, Shuming Shi, Yunbo Cao, and Hang Li. Title extraction from bodies of html documents and its application to web page retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 250–257, New York, NY, USA, 2005. ACM Press. (Cited on page 16.)

[69] Yasuto Ishitani. Document image analysis with cooperative interaction between layout analysis and logical structure analysis. In *Document Layout Interpretation and its Applications (DLIA)*, 1999. (Cited on page 11.)

[70] Yasuto Ishitani. Logical structure analysis of document images based on emergent computation. In *ICDAR*, pages 189–192, 1999. (Cited on pages 11 and 30.)

[71] K. Iwane, M. Yamaoka, and O. Iwaki. A functional classification approach to layout analysis of document images. In *Proc. of 2nd ICDAR*, pages 778–781, 1993. (Cited on page 12.)

[72] Anil K. Jain and Bin Yu. Document representation and its application to page decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):294–308, March 1998. (Cited on page 27.)

[73] Katharina Kaiser and Silvia Miksch. Information extraction. a survey. Technical Report Asgaard-TR-2005-6, Vienna University of Technology, Institute of Software Technology & Interactive Systems, May 2005. (Cited on page 6.)

[74] Tapas Kanungo and Song Mao. Stochastic language models for style-directed physical layout analysis of documents. *IEEE Transactions on Image Processing*, 12(5):583–596, May 2003. (Cited on page 12.)

[75] Thomas G. Kieninger. Table structure recognition based on robust block segmentation. In Daniel P. Lopresti and Jiangying Zhou, editors, *Proc. SPIE Document Recognition V*, volume 3305 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 22–32, April 1998. (Cited on pages 9 and 26.)

[76] K. Kise, M. Iwata, and K. Matsumoto. On the application of voronoi diagrams to page segmentation. In *Proc. of the Workshop on Document Layout Interpretation and Its Applications*, number IV-C, pages 1–4, September 1999. (Cited on pages 23 and 82.)

[77] K. Kise, O. Yanagida, and S. Takamatsu. Page segmentation based on thinning of background. In *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276*, page 788, Washington, DC, USA, 1996. IEEE Computer Society. (Cited on pages 13 and 23.)

[78] Koichi Kise, Akinori Sato, and Motoi Iwata. Segmentation of page images using the area voronoi diagram. *Comput. Vis. Image Underst.*, 70(3):370–382, 1998. (Cited on page 13.)

[79] Stefan Klink, Andreas Dengel, and Thomas Kieninger. Document structure analysis based on layout and textual features. In *Proceedings of the 4th IAPR International Workshop on Document Analysis Systems (DAS 2000)*, pages 99–111, December 2000. (Cited on page 12.)

[80] Markus Knauff, Reinhold Rauh, Christoph Schlieder, and Gerhard Strube. Mental models in spatial reasoning. In Christian Freksa, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, volume 1404 of *Lecture Notes in Computer Science*, pages 267–292. Springer, 1998. (Cited on pages 54 and 83.)

[81] David Konopnicki and Oded Shmueli. W3qs: A query system for the world-wide web. In *Proc. 21th VLDB*, pages 54–65, San Francisco, CA, USA, September 1995. Morgan Kaufmann Publishers Inc. (Cited on page 13.)

[82] Milos Kovacevic, Michelangelo Dilligenti, Marco Gori, and Veljko M. Milutinovic. Visual adjacency multigraphs - a novel approach for a web page classification. In *ECML/PKDD 2004: Proceedings of the Workshop W1 on Statistical Approaches to Web Mining (SAWM)*, pages 38–49, Pisa, Italy, September 2004. (Cited on pages 27, 28 and 82.)

[83] Joachim Kreich. Robust recognition of documents. In *Proc. of the 2nd ICDAR*, pages 444 – 447, October 1993. (Cited on page 11.)

[84] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(7):737–747, July 1993. (Cited on page 10.)

[85] Nickolas Kushmerick, Daniel S. Weld, and Robert B. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence*, pages 729–737, 1997. Chairperson-Daniel S. Weld. (Cited on page 13.)

[86] Laks V. S. Lakshmanan, Iyer N. Subramanian, and Fereidoon Sadri. A declarative language for querying and restructuring the web. In *RIDE '96: Proceedings of the 6th International Workshop on Research Issues in Data Engineering (RIDE '96) Interoperability of Nontraditional Database Systems*, page 12, Washington, DC, USA, 1996. IEEE Computer Society. (Cited on page 13.)

[87] D.X. Le and G.R. Thoma. Automated portrait/landscape mode detection on a binary image. In F. O. Huck and R. D. Juday, editors, *Proc. of SPIE, Visual Information Processing II*, volume 1961 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 202–212, August 1993. (Cited on page 20.)

[88] F. Lebourgeois, Z. Bublinski, and H. Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *Proc. of the 11th ICPR*, volume 2, pages 272–276, August, September 1992. (Cited on pages 10 and 18.)

[89] J. Liang, I.T. Phillips, and R. Haralick. A unified methodology for document structure analysis,. In *Document Layout Interpretation and its Applications (DLIA)*, 1999. (Cited on page 9.)

[90] S. Lim and Y. Ng. Extracting structures of html documents using a high-level stack machine. In *Proceedings of the 12 International Conference on Information Networking ICOIN*, Tokyo, Japan, 1998. (Cited on page 14.)

[91] Seung Jin Lim and Yiu-Kai Ng. A heuristic approach for converting html documents to xml documents. In *CL '00: Proceedings of the First International Conference on Computational Logic*, pages 1182–1196, London, UK, 2000. Springer-Verlag. (Cited on page 14.)

[92] Bing Liu, Robert Grossman, and Yanhong Zhai. Mining data records in web pages. In *9th. ACM SIGKDD*, pages 601–606, New York, NY, USA, August 2003. ACM Press. (Cited on page 15.)

[93] J. Liu, Y. Y. Tang, Q. He, and C. Y. Suen. Adaptive document segmentation and geometric relation labeling: Algorithms and experimental results. In *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276*, page 763, Washington, DC, USA, 1996. IEEE Computer Society. (Cited on page 9.)

[94] Wei Liu, Xiaofeng Meng, and Weiyi Meng. Vision-based web data records extraction. In *9th. WebDB*, pages 20–25, June 2006. (Cited on pages 5 and 16.)

[95] William S. Lovegrove and David F. Brailsford. Document analysis of pdf files: methods, results and implications. *Electronic Publishing, Origination, Dissemination and Design*, 8(3):207–220, June & September 1996. (Cited on page 9.)

[96] Donato Malerba, Floriana Esposito, and Oronzo Altamura. Adaptive layout analysis of document images. In *ISMIS '02: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, pages 526–534, London, UK, 2002. Springer-Verlag. (Cited on page 12.)

[97] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. Document structure analysis algorithms: A literature survey. In *Proc. SPIE Electronic Imaging, Document Recognition and Retreval X*, volume 5010, pages 197–207, January 2003. (Cited on page 6.)

[98] Simone Marinai, Emanuele Marino, and Giovanni Soda. Layout based document image retrieval by means of xy tree reduction. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 432–436, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on page 21.)

[99] Alberto O. Mendelzon, George A. Mihaila, and Tova Milo. Querying the world wide web. In *DIS '96: Proceedings of the fourth international conference on on Parallel and distributed information systems*, pages 80–91, Washington, DC, USA, December 1996. IEEE Computer Society. (Cited on page 13.)

[100] Jean-Luc Meunier. Optimized xy-cut for determining a page reading order. In *ICDAR*, pages 347–351. IEEE Computer Society, 2005. (Cited on pages 26, 30, 31, 47 and 82.)

[101] Phillip E. Mitchell and Hong Yan. Document page segmentation and layout analysis using soft ordering. *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, 01:1458, September 2000. (Cited on pages 9, 30 and 56.)

[102] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997. (Cited on pages 50 and 51.)

[103] Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):93–114, 2001. (Cited on page 14.)

[104] Morton Nadler. Document segmentation and coding techniques. *Computer Vision, Graphics, and Image Processing*, 28(2):240–262, 1984. (Cited on page 6.)

[105] George Nagy and S. Seth. Hierarchical representation of optical scanned documents. In *7th. Int. Conf. on Pattern Recognition*, pages 347–349, 1984. (Cited on pages 6, 7, 21 and 22.)

[106] George Nagy, Sharad C. Seth, and Spotswood D. Stoddard. Document analysis with an expert system. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice II (Proc. Int. Workshop, Amsterdam, the Netherlands, 19-21 June, 1985)*, pages 149–159. Elsevier Science, 1986. (Cited on pages 6 and 22.)

[107] D. Niyogi and S. N. Srihari. Knowledge-based derivation of document logical structure. In *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, page 472, Washington, DC, USA, 1995. IEEE Computer Society. (Cited on page 11.)

[108] N. Normand and C. Viard-Gaudin. A background based adaptive page segmentation algorithm. In *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, page 138, Washington, DC, USA, 1995. IEEE Computer Society. (Cited on pages 8, 18, 19, 25 and 82.)

[109] Lawrence O'Gorman. The document spectrum for page layout analysis. In Lawrence O'Gorman and Rangachar Kasturi, editors, *Document image analysis*, pages 214–225. IEEE Computer Society Press, Los Alamitos, CA, USA, 1993. (Cited on pages 12 and 22.)

[110] Alberto Pan, Juan Raposo, Manuel Álvarez, Paula Montoto, Vicente Orjales, Justo Hidalgo, Lucía Ardao, Anastasio Molano, and Ángel Viña. The denodo data integration platform. In *VLDB*, pages 986–989. Morgan Kaufmann, 2002. (Cited on page 14.)

[111] Theo Pavlidis and Jiangying Zhou. Page segmentation and classification. *CVGIP: Graph. Models Image Process.*, 54(6):484–496, 1992. (Cited on page 7.)

[112] Bernhard Pollak and Wolfgang Gatterbauer. Creating permanent test collections of web pages for information extraction research. In *Proc. 33rd SOFSEM, Vol.2*, pages 103–115, January 2007. (Cited on pages 5, 32, 35 and 36.)

[113] Dmitri Popov. Turn firefox into an archiving and research tool with scrapbook. *Linux-Magazine*, 73:84–86, December 2006. See http://www.linux-magazine.com/issues/2006/73/workspace. (Cited on page 33.)

[114] V. Shiv Naga Prasad and Justin Domke. Gabor filter visualization. Student Semester Project, Course 838 Information Visualization, University of Maryland, 2005. (Cited on pages 20, 21 and 82.)

[115] Dorian Pyle. *Data preparation for data mining*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. (Cited on page 50.)

[116] Xiaoguang Qi and Brian D. Davison. Web page classification: Features and algorithms. Technical Report LU-CSE-07-010, Dept. of Computer Science and Engineering, Lehigh University,, June 2007. (Cited on page 32.)

[117] Yu-Long Qiao, Meng Li, Zhe-Ming Lu, and Sheng-He Sun. Gabor filter based text extraction from digital document images. In *IIH-MSP*, pages 297–300. IEEE Computer Society, 2006. (Cited on pages 21 and 82.)

[118] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. (Cited on page 49.)

[119] R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986. (Cited on page 49.)

[120] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proc. of 3rd KR*, pages 165–176, 1992. (Cited on page 29.)

[121] Jochen Renz, Reinhold Rauh, and Markus Knauff. Towards cognitive adequacy of topological spatial relations. In *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, pages 184–197, London, UK, 2000. Springer-Verlag. (Cited on page 55.)

[122] Daniela Rus and Kristen Summers. Using white space for automated document structuring. Technical report, Cornell University, Ithaca, NY, USA, 1994. (Cited on page 11.)

[123] T. Saitoh, T. Yamaai, and M. Tachikawa. Document image segmentation and layout analysis. *Transactions Institute Elec. Info. and Comm. Eng.*, Info Sys 77(7):778–784, 1994. (Cited on page 8.)

[124] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. (Cited on page 21.)

[125] J. Sauvola and M. Pietikäinen. Page segmentation and classification using fast feature extraction and connectivity analysis. In *ICDAR '95: Proceedings of the Third International Conference on*

Document Analysis and Recognition, volume 02, page 1127, Washington, DC, USA, 1995. IEEE Computer Society. (Cited on page 9.)

[126] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 & 623–656, 1948. (Cited on pages 50 and 83.)

[127] Zhixin Shi and Venu Govindaraju. Multi-scale techniques for document page segmentation. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 1020–1024, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 9, 19 and 82.)

[128] Kai Simon and Georg Lausen. Viper: augmenting automatic information extraction with visual perceptions. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 381–388, Bremen, Germany, 2005. ACM Press. (Cited on page 16.)

[129] R. Sivaramakrishnan, I. T. Phillips, J. Ha, S. Subramanium, and R. M. Haralick. Zone classification in a document using the method of feature vector generation. In *3rd ICDAR, Vol.2*, page 541, Washington, DC, USA, August 1995. IEEE Computer Society. (Cited on pages 8 and 9.)

[130] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Mach. Learn.*, 34(1-3):233–272, 1999. (Cited on page 14.)

[131] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning block importance models for web pages. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 203–211, New York, NY, USA, May 2004. ACM Press. (Cited on page 53.)

[132] Lawrence A. Spitz. Style directed document recognition. In *Proc. of the 1st ICDAR*, pages 611–619, 1991. (Cited on page 7.)

[133] Lawrence A. Spitz. Style directed document recognition. In *IAPR Workshop on Document Layout Interpretation and its Application DLIA*, September 1999. (Cited on page 7.)

[134] Lawrence A. Spitz. Style-directed document segmentation. In *Symposium on Document Image Understanding Technology*, pages 195–199, 2001. (Cited on page 7.)

[135] Kristen Summers. *Automatic Discovery Of Logical Document Structures*. PhD thesis, Cornell University, Ithaca, NY, USA, August 1998. (Cited on pages 6 and 40.)

[136] Don Sylwester. Adaptive segmentation of document images. In *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 827, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 22.)

[137] Yoshitake Tsuji. Document image analysis for generating syntactic strcuture description. In *9th International Conference on Pattern Recognition*, volume 2, pages 744–747, November 1988. (Cited on page 7.)

[138] International Telecommunication Union. T.411: Information technology - open document architecture (oda) and interchange format: Introduction and general principles. http://www.itu.int/rec/T-REC-T.411/en, March 1993. (Cited on page 7.)

[139] Boulos Waked. Page segmentation and identification in document image analysis. Master's thesis, Concordia University, Montreal, September 2001. (Cited on page 24.)

[140] Dacheng Wang and Sargur N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47(3):327–352, 1989. (Cited on pages 7, 22 and 82.)

[141] Shin-Ywan Wang and T. Yagasaki. Block selection: a method for segmenting a page image of various editing styles. *icdar*, 01:128–133, August 1995. (Cited on page 8.)

[142] Yalin Wang and Jianying Hu. A machine learning based approach for table detection on the web. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 242–250, New York, NY, USA, May 2002. ACM Press. (Cited on page 32.)

[143] Watanabe and Sobue. Layout analysis of complex documents. In *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, page 4447, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on pages 12 and 22.)

[144] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005. (Cited on pages 49 and 50.)

[145] Kwan Y. Wong, Richard G. Casey, and Friedrich M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, November 1982. (Cited on pages 6, 17, 18 and 82.)

[146] Yewei Xue, Yunhua Hu, Guomao Xin, Ruihua Song, Shuming Shi, Yunbo Cao, Chin-Yew Lin, and Hang Li. Web page title extraction and its application. *Inf. Process. Manage.*, 43(5):1332–1347, September 2007. (Cited on page 16.)

[147] Yudong Yang and HongJiang Zhang. Html page analysis based on visual cues. In *6th ICDAR*, pages 859–864, Washington, DC, USA, September 2001. IEEE. (Cited on page 15.)

[148] Zhen Zhang, Bin He, and Kevin Chen-Chuan Chang. Understanding web query interfaces: best-effort parsing with hidden syntax. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 107–118, Paris, France, 2004. ACM Press. (Cited on pages 15, 29 and 82.)

[149] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. Fully automatic wrapper generation for search engines. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 66–75, Chiba, Japan, 2005. ACM Press. (Cited on page 16.)

[150] Hongkun Zhao, Weiyi Meng, and Clement Yu. Automatic extraction of dynamic record sections from search engine result pages. In *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 989–1000. VLDB Endowment, September 2006. (Cited on page 16.)

[151] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. Simultaneous record detection and attribute labeling in web data extraction. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 494–503, New York, NY, USA, 2006. ACM Press. (Cited on page 16.)

[152] Jun Zhu, Bo Zhang, Zaiqing Nie, Ji-Rong Wen, and Hsiao-Wuen Hon. Webpage understanding: an integrated approach. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 903–912, New York, NY, USA, August 2007. ACM Press. (Cited on page 16.)

[153] Jie Zou, Daniel Le, and George R. Thoma. Combining dom tree and geometric layout analysis for online medical journal article segmentation. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 119–128, New York, NY, USA, June 2006. ACM Press. (Cited on page 4.)