Machine Learning for Document Structure Recognition

Gerhard Paaß Iuliu Konya

June 22, 2009

Contents

1 Introduction

In the last years, there has been a rising interest in the easy access of printed material in large-scale projects such as Google Book Search Vincent [2007] or the Million Book Project Sankar *et al.* [2006]. To make this material amenable to browsing and retrieval the logical structure of documents into titles, headings, sections, and thematically coherent parts has to be recognized. To cope with large collections this task has to be performed in an automatic way. The result produced by a document understanding system, given a text representation, should be a complete representation of the document's logical structure, ranging from semantically high-level components to the lowest level components.

Document structure recognition can exploit two sources of information. On the one hand the layout of text on the printed page often gives many clues about the relation of different structural units like headings, body text, references, figures, etc.. On the other hand the wording and the contents itself can be exploited to recognize the interrelation and semantics of text passages.

There currently exist a wide range of algorithms specialized for certain parts of document analysis. In large scale applications these approaches have to cope with the vast variety of printed document layouts. A recent comparison is given by Shafait *et al.* [2006] showing that no single algorithm is uniformly optimal. As argued in Baird and Casey [2006], versatility is the key requirement for successful document analysis systems. Even for the same publisher, the layout of its publications changes drastically over time. This is especially visible when dealing with publications spanning over many decades or even centuries. As a general rule, more recently printed documents are also more complex, and the difference between the layouts used by different publishers becomes more pronounced. Thus it is extremely difficult to have algorithms consistently delivering good results over the whole range of documents.

Machine learning approaches are a potential remedy in this situation. Starting form a training set of structures documents they are able to extract a large number of features relevant for document structure. In contrast to manually built rule systems they are able to weight these features and the change of a few features does not lead to drastic loss of performance.

In this chapter we first discuss some advance approaches for detecting the structure of text based on the sequence of text objects and layout features. This can be formulated as a classification problem which may be solved by discrimiative classifiers like the support vector machine. To take into account structural features these approaches may be enhanced by tree kernels, as shown in section ???. As an alternative we discuss Conditional Random Fileds, which describe the interrelation of structural states of text and are able to include a large number of dependent features. In the second part we describe an approach based on minimum spanning trees. It is able to cope with multiple columns and embedded commercials having a non-Manhattan layout and may be automatically adapted to the different layouts of each publisher. It has been used in large scale newspaper digitization projects.

1.1 Conditional Random Fields

1.1.1 Basic Model

Let us consider the problem that we want to identify title and author in the following text snippet

The new bestseller:
Tears of Love
by Paula Lowe

For simplicity we may write the words of the snippet including newlines and mark them with T if they belong to the title, by A if they belong to the authors, and by O otherwise. This gives the two vectors x of words and y of unknown states

y	O	O	O	O	T	T	T	O	O	A	A
x	The	new	bestseller	\n	**Tears**	**of**	**Love**	\n	by	Paula	Lowe

Text data in documents has two characteristics: first, statistical dependencies exist between the words, we wish to model, and second, each word often has a rich set of features that can aid classification. For example, when identifying the title in documents we can exploit the format and font

properties of the title itself, but the location and properties of an author and an abstract in the neighborhood can improve performance.

To infer the unknown states we represent the relation between sequences y and x by a conditional probability distribution $p(y|x)$. More specifically let the variables $y = (y_1, \ldots, y_n)$ represent the labels of the word that we wish to predict with a set Y of possible values. Let the input variables $x = (x_1, \ldots, x_n)$ represent the observed words and their properties. If $I = \{1, \ldots, n\}$ is the set of indices of y then we denote the subvector corresponding to the indices in $A \subset I$ by y_A. Let $\phi_A(x, y_A) > 0$ be a *factor function* with x and the subvectors y_A as arguments and let $\mathcal{C}$ be a set of subsets of $A \subset I$. Each $\phi_A(x, y_A)$ is a function taking into account the relation between the labels in the subvector y_A, which often are the adjacent labels in the sequence. Then we represent the conditional distribution by a product of factor functions

$$p(y|x) = \frac{1}{Z(x)} \prod_{A \in \mathcal{C}} \phi_A(x, y_A) \tag{1}$$

Here $Z(x) = \sum_y \prod_{A \in \mathcal{C}} \phi_A(x, y_A)$ is a factor normalizing the sum of probabilities to 1.

The product structure enforces a specific dependency structure of the variables y_i. Consider the conditional distribution of y_i given all other variables. It may be written as

$$p(y_i|y_{D(i)}, x) = \frac{p(y_i, y_{D(i)}, x)}{\sum_{y_i \in Y} p(y_i, y_{D(i)}, x)} = \frac{\prod_{B \in \mathcal{C}, i \in B} \phi_B(x, y_B)}{\sum_{y_i \in Y} \prod_{B \in \mathcal{C}, i \in B} \phi_B(x, y_B)} \tag{2}$$

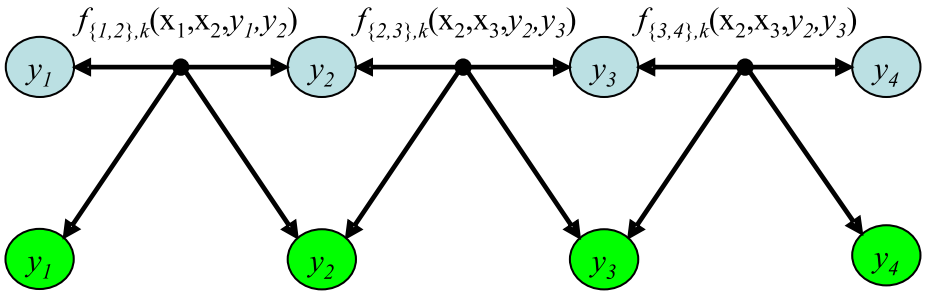
as the factor functions $\phi_A(x, y_A)$ where $i \notin A$ cancel. Therefore the conditional probability of y_i is completely determined if the values of x and the y_B are known for all B which contain i. The factor functions $\phi_A(x, y_A)$ describe the *interactions* between the argument variables. Obviously C determines the dependency structure of the components of y. A probability distribution of this form is called *conditional random field* (CRF) Lafferty *et al.* [2001] Sutton and McCallum [2007]. As dependencies among the input variables x do not need to be explicitly represented, rich, global input features x may be used. For example, in natural language tasks, useful features include neighboring words and word bigrams, prefixes and suffixes, capitalization, membership in domain-specific lexicons, and semantic information from sources such as WordNet.

Recently there has been an explosion of interest in CRFs, with successful applications including text processing [Taskar et al., 2002, Peng and McCallum, 2004, Settles, 2005, Sha and Pereira, 2003], bioinformatics [Sato and Sakakibara, 2005, Liu et al., 2005], and computer vision [He et al., 2004, Kumar and Hebert, 2003].

Usually there exists a number of different *features* for the same variables x, y_A. For $A = \{i\}$ for instance $\phi_A(x, y_i)$ may cover the feature that word x_i is in bold and $y_i = T$, i.e. is a title word. If we have K_A features for A then we may write $\phi_A(x, y_A) = \exp(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A))$. Here $\lambda_{A,k}$ is a real-valued *parameter* determining the importance of the real-valued *feature function* $f_{A,k}(x, y_A)$. The exponentiation ensures that the factor functions are positive. This yields the representation

$$p(y|x) = \frac{1}{Z(x)} \prod_{A \in \mathcal{C}} \exp\left(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A)\right) = \frac{1}{Z(x)} \exp\left(\sum_{A \in \mathcal{C}} \sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A)\right) \tag{3}$$

Often the feature functions are binary with value $f_{A,k}(x, y_A) = 1$ if the feature is present and $f_{A,k}(x, y_A) = 0$ otherwise. If $\lambda_{A,k} = 0$ the corresponding feature has no influence. For non-negative feature functions positive values for $\lambda_{A,k}$ indicate that the feature increases $p(y_A|x)$, while negative



values decrease the conditional probability. and have to be estimated from training data by maximum likelihood.

A a common special case is the *linear chain conditional random field*, where only interactions between y_t and y_{t-1} are allowed. If in addition we only take into account the corresponding inputs x_t and x_{t-1} the feature functions have the form. $f_{\{t-1,t\},k}(x_{t-1}, x_t, y_{t-1}, y_t)$. Therefore only the adjacent states y_{t-1} and y_t influence each other directly. The following figure shows such a linear chain with four states. For simplicity only a single type of feature function is shown.

Often it can be assumed, that the parameters do not depend on the particular t and hence $\lambda_{\{t-1,t\},k} = \lambda_{\{t,t+1\},k}$ for all t. This *parameter tying* drasticaly reduceds the number of unkown parameters. More general we may partition $\mathcal{C} = \{C_1, \ldots, C_Q\}$ where each C_q is a set of all A whose parameters are tied. Then we get the representation

$$p(y|x;\lambda) = \frac{1}{Z(x)} \exp\left(\sum_{C_p \in \mathcal{C}} \sum_{A \in C_p} \sum_{k=1}^{K_A} \lambda_{p,k} f_{A,k}(x, y_A)\right) \tag{4}$$

We may estimate the unkown parameters according to the maximum likelihood criterion. Assume we have observed a number of i.i.d observations $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$, e.g different documents which are already labeled with the states. Differentiating the log-likelihood function $\ell(\lambda) = \log \prod_n p(y^{(n)}|x^{(n)};\lambda)$ with respect to $\lambda_{p,k}$ yields

$$\frac{\partial \ell(\lambda)}{\partial \lambda_{p,k}} = \sum_{n=1}^{N} \left[\sum_{A \in C_p} f_{A,k}(x^{(n)}, y_A^{(n)}) - \sum_{A \in C_p} \sum_{y_A \in Y_A} p(y_A|x^{(n)};\lambda) f_{A,k}(x^{(n)}, y_A)\right]$$

where Y_A is the set of all possible y_A and $p(y_A|x^{(n)};\lambda)$ is the probability of y_A given $x^{(n)}$ and the current parameter values λ.

The first sum contains the observed feature values for $f_{A,k}(x^{(n)}, y_A^{(n)})$ and the second sum consists of the expected feature values given the current parameter λ. If the gradient is zero both terms have to be equal. It can be shown that the log likelihoodfunction is concave and hence may be efficiently maximized by second-order techniques such as conjugate gradient or L-BFGS. To improve generalization a quadratic penalty term may be added which keeps the parameter values small.

Gradient training requires the computation of the marginal distributions $p(y_A|x^{(i)})$. In the case of a linear chain CRF this can efficiently done by the forward-backward algorithm requiring 2*N steps. For tree-structured networks we may use the ..., which. Networks with cycles require more effort as

the exact computation grows exponentially with the diameter: loopy belief propagation (see section 1.1.4).

If the parameters are known we have to determine the most likely state configuration for a new input $x^+ = (x_1^+, \ldots, x_n^+)$

$$y^* = \arg\max_y p(y|x^+; \lambda)$$

which in the case of linear chain models can be efficiently calculated by dynamic programming using the Viterbi algorithm. During prediction the linear-chain CRF takes into account the correlations between adjacent states, which for many problems increase the prediction quality. Other problems requiring long-range correlations between states are described in sections 1.1.3 and 1.1.4.

1.1.2 Application of Linear-Chain CRFs to Structure Information extraction

Peng and McCallum [2004] applied linear chain CRFs to the extraction of structural information from scientific research papers. In their header extraction task they consider the first part of a paper which has to be labeled with the following states: title, author, affiliation, address, note, email, date, abstract, introduction, phone, keywords, web, degree, publication number, and page. A second reference task labels the references at te end of a paper with the following states: author, title, editor, booktitle, date, journal, volume, tech, institution, pages, location, publisher, note. They used the following features:

- Local features describing the curret word x_i: word itself, starts with capital letter, only capital letters, contains digit, only digits, contains dot, contains "-", acronym, capital letter and dot, matches regular expressions for phone number, zipcode, URL, email.
- layout features: word at begin of line, word in the middle of line, word at the end of a line.
- External lexicon features: Word in author list, word in date list (e.g. Jan. Feb.), word in notes.

On a training set with 500 headers they achieve an average F1 of 94% for the different fields, compared to 90% for SVMs and 76% for HMMs. For the reference extraction task trained on 500 articles they yield and F1-value of 91.5% compared to 77.6% for an HMM. They found that the Gaussian prior consistently performs best.

Schneider [2006] uses linear CFRs to extract information like conference names, titles, dates, locations, and submission deadlines from call for papers with the goal to compile conference calenders automatically. He models the sequence of words in a CFP and uses the following layout features: first / last token in the line, first/last line in the text, line contains only blanks / punctuations, line is indented, in first 10 / 20 lines of the text. Using a training dataset of 128 CFPs they achieve an average F1-value of about 60-70% for the title, date and other fields of a CFP. More difficult is te identification of a the co-located main conference which has only an F1-value of 35%.

1.1.3 Discriminative Parsing Models

Document structure extraction problems can be solved more effectively by learning a discriminative context free grammar (CFG) from training data. According to Viola and Narasimhan [2005] Awasthi *et al.* [2007] a grammar has several distinct advantages: long range, even global, constraints can

be used to disambiguate entity labels; training data is used more efficiently; and a set of new more powerful features can be introduced. The specific problem Viola and Narasimhan [2005] consider is of extracting personal contact, or address, information from unstructured sources such as documents and emails.

A CFG consists of a set of terminals $\mathcal{T} = \{w_1, \ldots, w_V\}$ and a set of non-terminals $\mathcal{N} = \{N_1, \ldots, N_n\}$, a designated start symbol N_1 and a set of rules or productions $\mathcal{R} = \{R_i : N_{j_i} \rightarrow \zeta_i\}$ where ζ_i is a sequence of terminals and non-terminals in $\mathcal{N} \cup \mathcal{T}$. A parse tree for a sequence $w_1, \ldots, w_m$ is a tree with $w_1, \ldots, w_m$ as leaf nodes and some $N_i \in \mathcal{N}$ as interior nodes such that the child nodes of an interior node are generated by a rule $R_i \in \mathcal{R}$. Associated with each rule is a score $S(R_i)$. The score of a complete parse tree is the sum of all scores of the rules used in the parse tree. The CKY (Cook-Kasami-Younger) algorithm (see Manning and Schütze [1999]) can compute the parse with the highest score in time $O(n^3 \cdot |\mathcal{R}|)$, which is feasible for relatively small m.

Assume that a nonterminal N_{j_i} generates the terminals $w_a, \ldots, w_b$ then the probability of a rule may be written by a loglinear expression

$$p(R_i) = \frac{1}{Z(\lambda(R_i), a, b, R_i)} \exp \sum_{k=1}^{F} \lambda_k(R_i) f_k(w_1, \ldots, w_m, a, b, N_{j_i} \rightarrow \zeta_i)$$

Here N_{j_i} directly or indirectly generates $w_a, \ldots, w_b$ and $f_1, \ldots, f_k$ is the set of features similar to the CRF features above, which may depend on all terms in the parenthesis. In principle, these features are more powerful than the linear-chain CRF-features because they can analyze the sequence of words associated with the current non-terminal and not only for the direct neighboring words. $\lambda_k(R_i)$ is the weight of feature k for R_i and $Z(\cdot)$ is a factors ensuring that the probabilities add up to 1.

As for the CRF this loglinear model is not intended to describe the generative process for $w_1, \ldots, w_m$ but aim at discriminating between different parses of $w_1, \ldots, w_m$. For training use a training set of documents manually labeled with the correct parse tree. They semiautomatically infer a set $\mathcal{R}$ of production rules and a set of features. The weights $\lambda_k(R_i)$ of the features for production rule R_i are determined by the perceptron learning algorithm, which successively increases weights for examples with active features and decreases weights for samples with inactive features.

The apply this approach to a CRF trained by the voted perceptron algoritm. They used a data set with about 1500 contact records with names adresses, etc. for training. For only 27% of the records in the training set an error occured, while the linear chain CRF had an error rate of 55%. This means that taking into account non-local information by the parse tree approach cut the error in half.

1.1.4 Graph-Structured Model

Up to now we have analyzed document structures with an inherent sequence of elements for the linear chain CRF or discriminative parsing models. We now discuss graph-like structures with more complex dependencies.

As an example consider the problem of processing newspaper archives. After scanning and applying OCR, low-level algorithms may be used to identify elements like lines, paragraphs, images, etc. The 2-dimensional page analysis can go further and establish spatial logical relationships between the elements, like ”touch”, "below", ”right of”, etc. Especially in newspapers with multi-column layout the

Figure 1: For multicolumn newspaper layout the relation between articles, paragraphs, images and tables is ambiguous and may be modeled by a probabilistic relational model.

sequence of paragraphs of an article in different columns or even on continuation pages is not unique. In the same way the assignment of tables, figures and images located somewhere on the page to an article is a challenging problem.

The low-level analysis generates a number of object o_i, e.g. lines, paragraphs, articles, images, etc. For a some pairs of these objects relations r_j may be specified, e.g. *image* left-of *article* , *image* belongs-to *article, article* below *article.* Each object and each relation has an associated type $t(o_i)$ or $t(r_i)$. Depending on the type each object and each relation is characterized by type-specific attributes, e.g. topic, title, or x-y-position. This yields for each type t a type-specific attribute vector $x_{o_i}^{t(o_i)}$ for an object or and attribute vector $x_{r_i}^{t(r_i)}$ for a relation. The following figure shows a small example network of relations between articles and images of a newspaper. A probabilistic relational network (PRM) (Taskar *et al.* [2002], Getoor and Taskar [2007], Neville and Jensen [2007]) represents a joint distribution over the attributes $x_{o_i}^{t(o_i)}$ and $x_{r_i}^{t(r_i)}$ of objects and relations.

Attributes of an object or relation can depend probabilistically on other attributes of the same or other objects or relations. For example the probability of image belonging to an article is higher if it is located close to it. In the same way the probability of an image belonging to an article is higher if the topic of the caption and the topic of the article are similar. These dependencies can be exploited in a probabilistic relation model.

In a linear chain CRF we had a generic dependency template between the states of successive states in the chain. This resulted in using the same parameters independent of the step index or the specific sentence. In the same way probabilistic relational models may define a number generic dependency templates depending on the types of the involved items. This approach of typing items and tying parameters across items of the same type is an essential component for the efficient learning of PRMs. It enables generalization from a single instance by decomposing the relational graph into multiple examples of each item type (e.g., all image objects), and building a joint model of dependencies between and among attributes of each type.

The resulting probabilistic dependency network is a graph-structured CRF (4) where parameters are tied in a specific way. This model is discussed in depth in Sutton and McCallum [2007]. A number of variants of CRF models have been developed in recent years. Dynamic conditional random fields Sutton *et al.* [2004] are sequence models which allow multiple labels at each time step, rather than single labels as in linear-chain CRFs. Lumped label CRFs Paaß and Reichartz [2009]allow to include

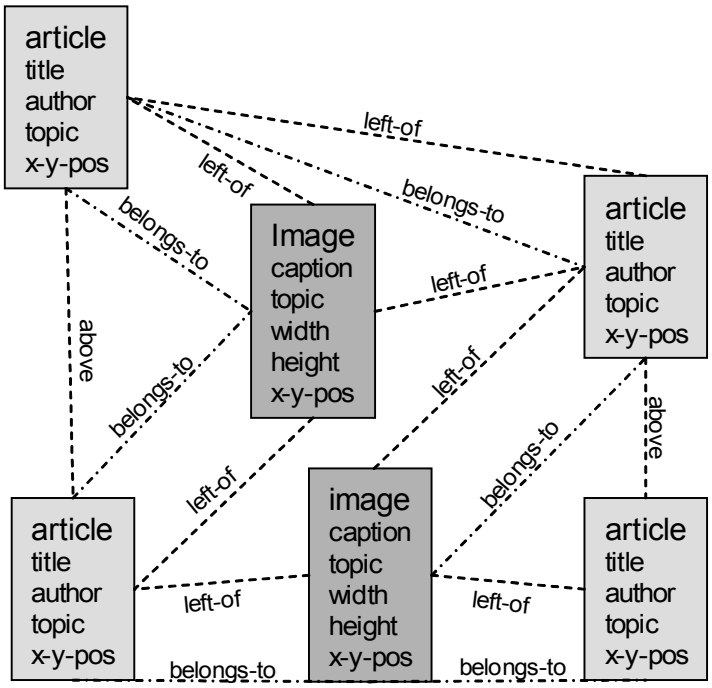


Figure 2: Articles and images of a newspaper page are characterized by a number of attributes. Between a subset of pairs different types of relations exist.

observations, where only a subset of labels is observed and it is known that one of the labels in the subset is the true label. Finally, Markov logic networks Richardson and Domingos [2006] are a type of probabilistic logic network in which there are parameters for each first-order rule in a knowledge base. These first-order rules may, for example, be exploited to specify constrainst between layout elements.

Parameter estimation for general CRFs is essentially the same as for linear-chains, except that computing the model expectations requires more general inference algorithms. Whenever the structure of the relationships between elements form an undirected graph, finding exact solutions require special graph transformations and eventually the enumeration of all possible annotations on the graph. This results in the exponential complexity of model training and inference. To make it tractable, several approximation techniques have been proposed for undirected graphs; these include variational and Markov Chain Monte Carlo methods.

A number of alternatives exist:

- Gibbs sampling Finkel *et al.* [2005], where for each training example the labels are selected randomly according to the conditional distribution (2). The required probabilities can be estimated from the resulting joint ditribution of labels.
- Loopy belief propagation Sutton and McCallum [2004], performing belief propagation, which is an exact inference algorithm for trees, ignoring part of the links.
- Pseudo-likelihood approaches Besag [1975] which instead of the predicted labels use the observed label values to predict a label from its environment.

Chidlovskii and Lecerf [2008] use a variant of probabilistic relational models to annalyze the structure of documents. They aim at annotating lines and pages in layout-oriented documents which correspond to the beginning of sections and section titles. While for a local network corresponding to linear chain CRFs they get an F1-value of 73.4% which is increased to 79.5% for a graph-structured probabilistic relational network.

There are other, more heuristic, models taking into account graph-structured dependencies. Wisniewski and Gallinari [2007] consider the problem of sequence labeling and propose a two steps method. First they use a local classifier for the initial assignment of elements without the taking into account dependencies. Then a relaxation process successively takes into account non-local dependecies to propagate information and ensure global consistency. They test their approach on a collection of 12000 course descriptions which have to be annotated with 17 different labels such as lecturer, title, start time or end time; each description contains between 4 and 552 elements to be extracted. For a CRF they report an F1-value of 78.7%, for a Probabilistic Context Free Grammar using maximum entropy estimators to estimate probabilities they yields 87.4% and the relaxation model arrives at an F1-value of 88.1%.

Nicolas: Document Image Segmentation Using a 2D Conditional Random Field Model ICDAR 2007

Handwriting: In this work we have proposed a Conditional Random Field model for 2D data labelling, in particular for document image segmentation. One of the main advantages of this model is that it can be learned automatically using machine learning procedures, so no manual parameter setting is necessary. This allows an easy adaptation to different types of documents and different analysis tasks. The results we have obtained on Flaubert's manuscripts show that the proposed model provides better results than MRF generative models. These results are similar to those presented in

2 Document Analysis for Large-Scale Processing

Despite intensive research in the area of document analysis, the research community is still far from the desired goal, a general method of processing images belonging to different document classes both accurately and automatically. We will see that while geometric layout analysis methods are fairly mature, logical layout analysis research is mainly focused on journal articles. The automatic discovery of logical document structure would enable a multitude of electronic document tools, including markup, hyperlinking, hierarchical browsing and component-based retrieval Summers [1995]. For this purpose, the application of machine learning techniques to arrive at a good solution has been identified by many researchers as being a promising new direction to take Marinai and Fujisawa [2008].

The current section is dedicated to the presentation of a rule-based module for performing logical layout analysis. The described module has been extensively used as part of an automatic system in the processing of large-scale (i.e. $>$100.000 pages) newspaper collections. As can be seen from figure 3, a generic DIU system must incorporate many specialized modules. Logical layout analysis is considered to be one of the most difficult areas in document processing and is a focal point of current research activity. We will also discuss the applicability of the previously described machine learning approaches as a replacement for the traditional rule-based methods. As a prelude, for the sake of completeness, a brief overview on the current research in geometric layout analysis is presented before going into the state-of-the-art algorithms for logical layout analysis.

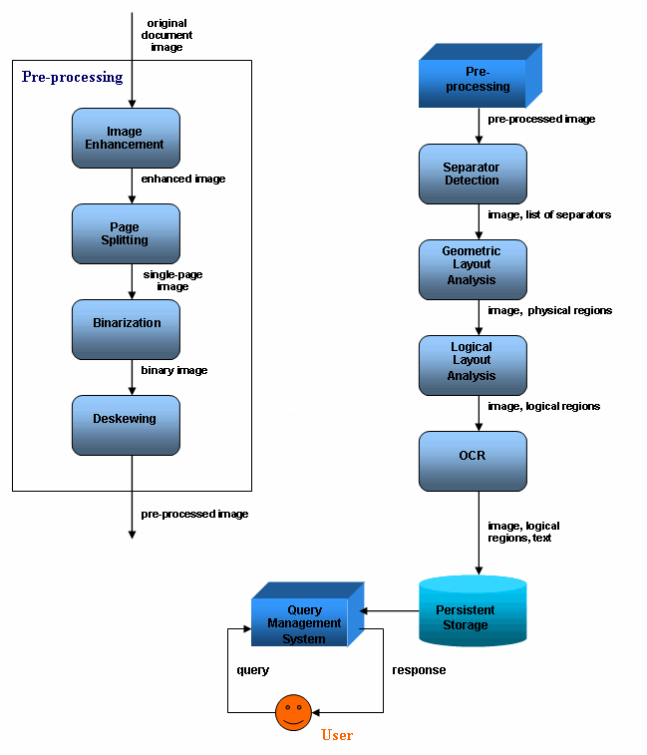
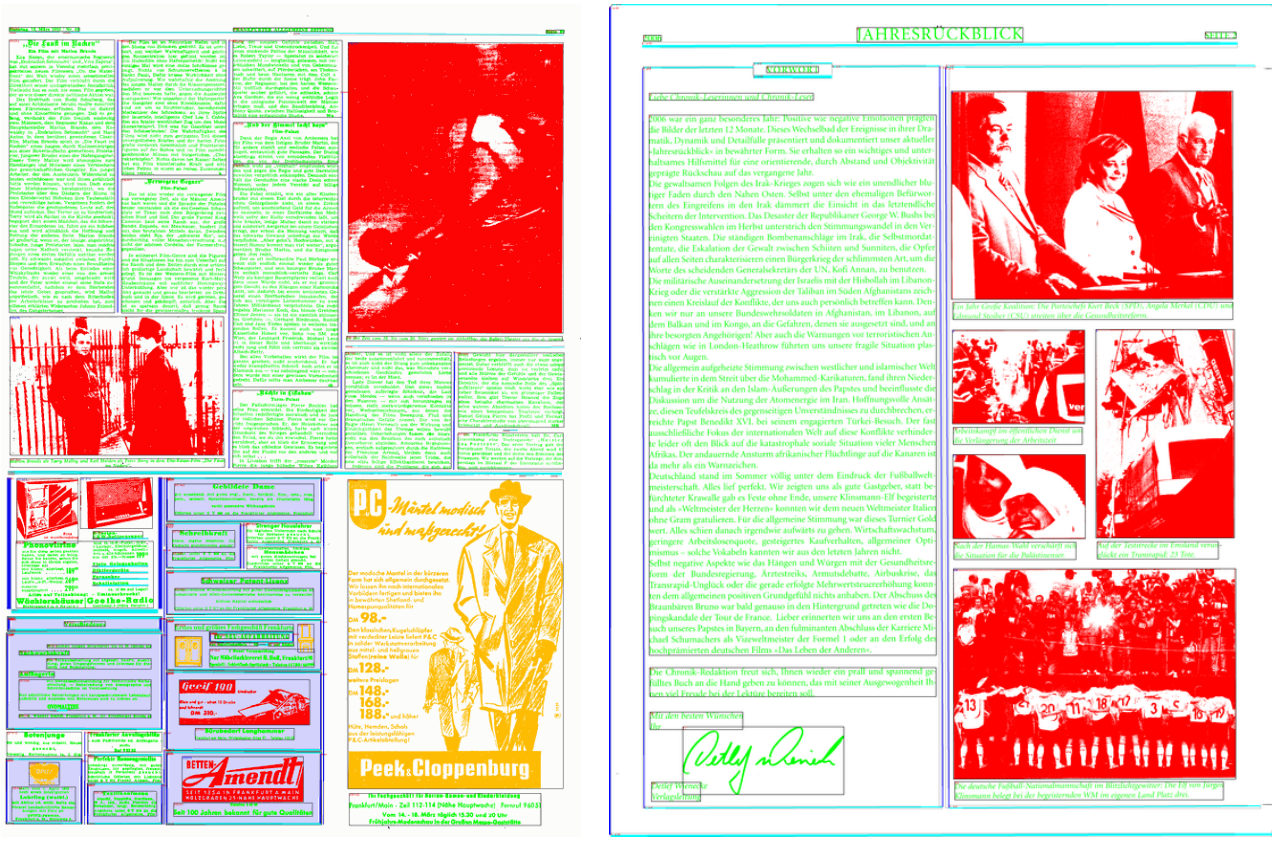


Figure 3: Functional model of a complete, generic DIU system. The ordering of some subsystems may vary, depending on the application area



a) b)

Figure 4: Example of page segmented images from: a) newspaper; b) chronicle. Color legend: green= text, red= image, orange= drawing, blue= vertical separator, cyan= horizontal separator, darkened background= frame box

2.1 State of the Art

2.1.1 Geometric Layout Analysis

The purpose of geometric layout analysis (or page segmentation) is to segment a document image into homogeneous zones, and to categorize each zone into a certain class of *physical layout elements*. Most commonly, the physical layout elements are divided into text, graphics, pictures, tables, horizontal and vertical rulers. It is important to note that in the specialized literature there exists no consensus on the number of physical classes considered, the number depending mostly on the target domain.

Ideally, the page segmentation process should be based solely on the geometric characteristics of the document image, without requiring any a priori information (such as a specific document type - e.g. newspaper, engineering drawing, envelope, web page). Many current page segmentation algorithms are able to meet this condition satisfactorily. In the vast majority of cases, however, the input image is assumed to be noise free, binary, and skew-free.

Traditionally, page segmentation methods are divided in three groups: *top-down* (model-driven), *bottom-up* (data-driven) and *hybrid* approaches. In top-down techniques, documents are recursively

divided from entire images to smaller regions. These techniques are generally very fast, but they are only useful when a priori knowledge about the document layout is available. To this class belong methods using projection profiles Ha *et al.* [1995], X-Y cuts Ha *et al.* [1995], or white streams Akindele and Belaid [1993]. Bottom-up methods start from pixels, merging them successively into higher-level regions, such as connected components, text lines, and text blocks. These methods are generally more flexible and tolerant to page skew (even multiple skew), but are also slower than top-down methods. Some popular bottom-up techniques make use of region growing (Jain and Yu [1998]; Kise *et al.* [1998]), run-length smearing Wahl *et al.* [1982], or mathematical morphology Gatos *et al.* [2005]. Many other methods exist which do not fit exactly into either of these categories; they were consequently called hybrid methods. Hybrid approaches try to combine the high speed of the top-down approaches with the robustness of the bottom-up approaches. Within this category fall all texture-based approaches, such as those employing Gabor filters, multi-scale wavelet analysis Doermann [1995], or fractal signatures Tang *et al.* [1995].

Many other algorithms for region detection have been proposed in the literature. For a more complete overview one may consult the most recent surveys and methods, such as Cattoni *et al.* [1998]; Mao *et al.* [2003]; Shafait *et al.* [2006]. Page segmentation methods are being evaluated from time to time, e.g. by Shafait *et al.* [2006] who compare the performance of six algorithms and, most recently, in the 2007 ICDAR competition described by Antonacopoulos *et al.* [2007]. As one may see from the results obtained in the recent years, current page segmentation algorithms perform quite well in the task of separating text and non-text regions. An evaluation of the page segmentation results produced by the module used in our DIU system on a set of 22 newspaper images coming from 6 different publishers has shown an accuracy of about 95% correctly separated text regions for the text-non-text separation task.

2.1.2 Logical Layout Analysis

The purpose of logical layout analysis is to segment the physical regions into meaningful *logical units* according to their type (e.g. text lines, paragraphs), assign a *logical label* to each of the determined regions (e.g. title, caption), as well as to determine the *logical relationships* between the logical regions (e.g. reading order, inclusion in the same article). Note that in case the processed document type is a periodical, logical layout analysis is also referred to as *article segmentation*.

The set of available logical labels is different for each type of document. For example: title, abstract, paragraph, section, table, figure and footnote are possible logical objects for technical papers, while: sender, receiver, date, body and signature emerge in letters. Logical relationships are typically represented in a hierarchy of objects, depending on the specific context Cattoni *et al.* [1998]. Examples of relations are cross references to different parts of an article or the (partial) reading order of some parts of a document. Taking into consideration all these aspects, it becomes clear that logical layout analysis can only be accomplished on the basis of some kind of a priori information (knowledge) about the document class and its typical layout, i.e. a model of the document. Such knowledge can be represented in very different forms (e.g. heuristic rules, formal grammars, probabilistic models such as Hidden Markov Models, a.s.o.). Cattoni *et al.* [1998] contains a survey of the different document formats used in modern document image understanding systems.

The number of available logical layout analysis algorithms is much lower than that of geometrical layout analysis algorithms, as the difficulty of the task is significantly higher. This section will only present the main ideas of a few methods and the interested reader is advised to consult one of the dedicated survey papers (e.g. Haralick [1994]; Cattoni *et al.* [1998]; Mao *et al.* [2003]).

Tsujimoto and Asada [1992] regarded both the physical layout and logical structure as trees. They transform the geometrical layout tree into a logical layout tree by using a small set of generic rules suitable for multi-column documents, such as technical journals and newspapers. The physical tree is constructed using block dominating rules. The blocks in the tree are then classified into head and body using rules related to the physical properties of the block. Once this logical tree is obtained, the final logical labels are assigned to the blocks using another set of rules. The logical labels considered are: title, abstract, sub-title, paragraph, header, footer, page number, and caption. A virtual field separator technique is introduced, in which separators and frames are considered as virtual physical blocks in the physical tree. This technique allows the tree transformation algorithm to function with a low number of transformation rules. The authors tested their algorithm on 106 pages from various sources and reported a logical structure recognition accuracy of 88.7%. Errors were due to inaccurate physical segmentation, insufficient transformation rules, and the fact that some pages did not actually have hierarchical physical and/or logical structures.

A general algorithm for automatic derivation of logical document structure from physical layout was described by Summers [1995]. The algorithm is divided into segmentation of text into zones and classification of these zones into logical components. The document logical structure is obtained by computing a distance measure between a physical segment and predefined prototypes. The set of properties assigned to each prototype are the parameters from which each distance value is calculated. The properties include contours, context, successor, height, symbols, and children. Basic textual information was also used in order to obtain a higher accuracy. The algorithm was tested on 196 pages from 9 randomly selected computer science technical reports. The labeling result of each text block was characterized as correct, over-generalized, or incorrect. Two metrics, precise accuracy and generalized accuracy, were used to evaluate the performance. Both average accuracy values were found to be greater than 86%.

Niyogi and Srihari [1995] presented a system called DeLoS for document logical structure derivation. In their system, the algorithm is regarded to be the result of applying a general rule-based control structure, as well as a hierarchical multi-level knowledge representation scheme. In this scheme, knowledge about the physical layouts and logical structures of various types of documents is encoded into a knowledge base. The system included three types of rules: knowledge rules, control rules, and strategy rules. The control rules control the application of knowledge rules, whereas the strategy rules determine the usage of control rules. A document image is first segmented using a bottom-up algorithm, followed by a geometric classification of the obtained regions. Finally, the physical regions are input into the DeLoS system and a logical tree structure is derived. The DeLoS system was tested on 44 newspaper pages. Performance results were reported in terms of block classification accuracy, block grouping accuracy, and read-order extraction accuracy.

In the recent years, research on logical layout analysis has shifted away from rigid rule-based methods toward the application of machine learning methods in order to deal with the required versatility. There are several examples for this. Esposito *et al.* [2004] employ machine learning in almost every

aspect of document analysis, from page segmentation to logical labeling. Their methods are based on inductive learning of knowledge that was hand-coded in previous approaches. Chen *et al.* [2007] use a set of training pages to learn specific layout styles and logical labels. An unknown page is recognized by matching the page's layout tree to the trained models and applying the appropriate zone labels from the best fit layout model. Similarly, the method of van Beusekom *et al.* [2007] finds for a given unlabeled page the best matching layout in a set of labeled example pages. The best match is used to transfer the logical labels to the unlabeled page. The authors see this as a light-weight yet effective approach. Rangoni and Belaïd [2006] use an artificial neural network as basis for their approach. Instead of a Multi Layer Perceptron where the internal state is unknown, they implement a Transparent Neural Network that allows introduction of knowledge into the internal layers. The approach features a feedback mechanism by which ambiguous results can be resolved by proposing likely and unlikely results to the input layer based on the knowledge about the current context. The input layer can respond by switching between different feature extraction algorithms, e.g. for determining the word count in a given block.

The logical layout analysis methods described so far have not been evaluated rigorously on layouts more complex than journal papers. The very complex newspaper layouts are for example the subject of Furmaniak [2007]. This is one of very few publications on the matter of article segmentation. It appears that this reflects the difficulty of the task. Yet, the author realizes that the mass digitalization of newspapers will be one of the next steps after the current wave of book digitalization projects. He proposes a method for learning the layout of different newspapers in an unsupervised manner. In a first stage, a word similarity analysis is performed for each pair of neighboring text blocks. The second stage uses geometric and morphological features of pairs of text blocks to learn the block relations that are characteristic for a specific newspaper layout. Results with high confidence from the word similarity analysis serve as ground truth for the training of the second stage. This method gives promising results and further strengthens the machine learning approach to logical layout analysis.

It is very important to note that in the area of logical layout analysis, there do not exist any standardized benchmarks or evaluation sets, not even algorithms for comparing the results of two different approaches. This is a gap that needs to be filled in future research, as principled evaluation is the only way to convincingly demonstrate advances in logical layout analysis research.

2.2 Minimum Spanning Tree-based Logical Layout Analysis

In case of *newspaper pages* or other *complex layout documents*, the logical layout analysis phase must be able to cope with multiple columns and embedded commercials having a non-Manhattan layout. Most importantly however, the approach has to be flexible enough so as to be readily adaptable (or adapt automatically) to the different layouts of each publisher. The current section contains the concise description of an article segmentation method, which, based on the construction of a minimum spanning tree (MST), is able to handle documents with a great variety of layouts.

Previous approaches using the MST in document layout analysis were proposed by Ittner and Baird [1993] and Dias [1996]. Ittner and Baird construct the MST from the centers of the connected components in the document image, and by means of a histogram of slopes of the tree edges, the authors are able to detect the dominant orientation of the text lines. Their method is based on the

assumption that inter-character distance is generally lower than inter-line spacing. The algorithm of Dias constructs the MST in a similar way and using the automatically determined inter-character (horizontal) and inter-line (vertical) spacing as splitting thresholds for the tree edges, it produces as output a segmentation of the document page into text regions. As input, the algorithm assumes that the input page is noise-free and contains only text, i.e. all non-text physical regions have been previously removed from the image via specialized filters. The most important problems observed by Dias are the sensitivity of the MST to noise components and the fact that a single incorrectly split branch can potentially produce a poor segmentation.

The MST-based algorithm introduced in this section requires as input a list containing the bounding boxes of all connected components belonging to text regions as well as the lists of vertical and horizontal separators detected in the given newspaper page. The first step consists of simply grouping the connected components into *text lines*. This can be accomplished by many algorithms, e.g. the geometric algorithm proposed in Jain and Yu [1998]. Several features are computed for each text line, the most important being the stroke width, the x height and the capital letter height for the font, the set of intersected layout columns and its position therein (e.g. left- or right-aligned or centered). Based on these features, one may now compute an optimal (i.e. minimize the total merging cost) set of *text regions* formed by vertically merging adjacent text lines having similar enough characteristics. This step can be accomplished by a dynamic programming approach. The costs of merging two vertically adjacent regions/lines is given by a measure of the similarity between their computed features. Note that here one may also include rules which take into account common formatting conventions, such as indentation at the beginning of each paragraph, in order to prevent text regions from spanning over several paragraphs. A single threshold is needed for this step, namely a stopping criterion for the merging of two regions. The threshold can be determined experimentally, and can subsequently be used for a wide range of publications, as shown by our experience.

At this point, it is possible to compute a *compound distance measure* between any two text regions as a weighted mean of the Euclidean distance between their bounding boxes and a value directly related to the "logical distance" between the two text blocks. The logical distance between two text blocks is asymmetrical and directly influenced by the number and type of separators present between the two text blocks, as well as by their feature similarity (as used for text region creation). The weights assigned to each of these components can and must be adjusted so as to match the different layouts used by a certain newspaper publisher. In order to be able to compute a meaningful logical distance between two blocks, we have additionally performed two steps before it: *detection of titles* and *detection of captions*. By using this additional information (which is in most cases relatively simple given a certain layout), one may compute more accurate logical distances between text blocks. For example a regular text block located before a title block in reading order will have a high logical distance to it (a probable article ending is located between them). A *hierarchy of titles* proved beneficial to use in our tests, as it allows the formulation of rules such as: a lower-level title located (in reading order) after a higher-level title with no other title in between has a low logical distance to it (as they are very likely part of the same article). By using the compound distance measure between text blocks, the MST of the document page can be constructed in the next step of the algorithm. It is important to notice that hereby the inherent noise sensitivity of the MST is significantly reduced, due to the usage of higher-order page components (i.e. logical blocks instead of just connected components). Next,

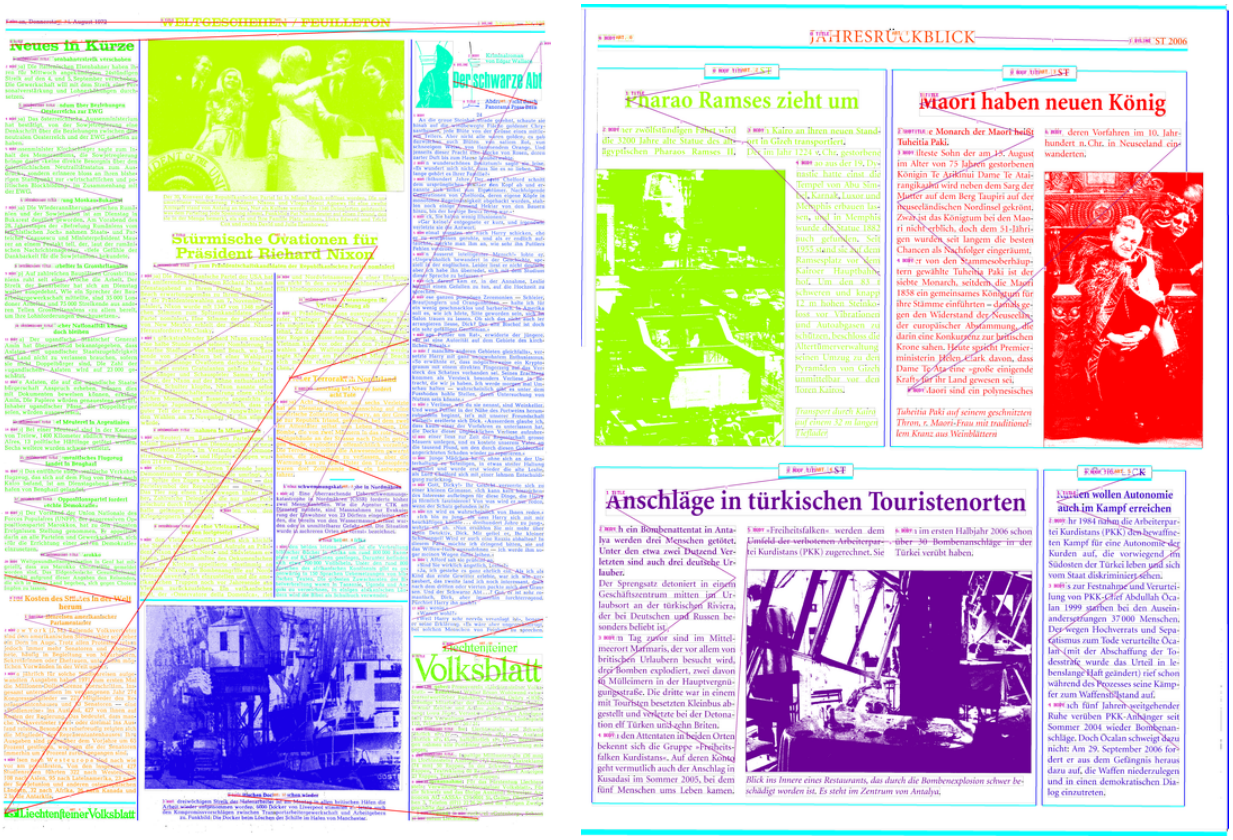
a) b)

Figure 5: Example of MST-based article segmentation on newspaper image: a) initial graph edges; b) MST result

the obtained tree is split into a set of smaller trees, each one ideally corresponding to an article. The splitting operation is done by removing the edges which have weights greater than a certain threshold value. The considered threshold value is closely related to the logical distance between blocks, and it should be adjusted according to the layout templates used by each publisher.

Finally, a suitable reading-order determination algorithm can be applied separately, for each article. The determination of the reading order is a hard problem and may depend not only on the geometric layout of a document (which varies widely among publishers even for the same document type), but also on linguistic and semantic content. For all our processing tasks, as well as all the described experiments we have used the method proposed by Breuel [2003], enriched with information regarding the layout columns present in the newspaper image. The method of Breuel uses solely geometric information about the text blocks (i.e. bounding boxes, vertical overlaps) and internally performs a *topological sort* given a list of pairs of text regions, each sorted in reading order.

A *post-processing stage* was found to be useful in many cases where an obtained tree actually corresponds to merely a part of an article. This is usually the case when text blocks not having an overlain title section are identified as independent articles. Such situations can readily be detected at this point, followed by a merge at the end of the previous article having a title (if such an article exists).



a) b)

Figure 6: Example of article segmented images from: a) newspaper; b) chronicle. Articles have distinct colors and the line segments indicate the detected reading order

The previous article can be found by searching backward (toward the head) in the list containing the articles sorted in reading order, for the first article which has no horizontal separator between itself and the considered article. This procedure has the advantage that it is independent of the specific language-dependent reading order algorithm previously employed.

The layout analysis algorithm described in this section has the advantage of being very fast, robust to noise and easily adaptable to a wide variety of document layouts. Its main shortcoming however, is the need to manually adapt the logical distance measures for each publisher or layout type. Also, the current algorithms does not need or take into account the text within each block, which may prove useful in case of more complex layouts. The future application of machine learning algorithms (such as those described in the previous sections of the chapter) for automating these tasks is a most promising approach.

2.2.1 Evaluation

All algorithms described in this section were incorporated in an in-house developed DIU system and successfully used for segmenting several large (>10.000 pages) newspaper collections. No formal

evaluation of the article segmentation results was performed on the respective collections, as a meaningful evaluation can only be performed by humans, which is of course prohibitive for such large quantities of data. However, a formal testing of our methods was done on 100 multi-column chronicle pages from the year 2006. Examples of the layout can be observed in figures 4 and 6. The original input images had 24-bit color depth and had a resolution of 400dpi (approx. 4000x5000 pixels). In these conditions, the total processing time for the article segmentation (incl. text line- and region detection and labeling of titles and captions) on one document image was about 8 seconds on a computer equipped with an Intel Core2Duo 2.66GHz processor and 2GB RAM. In the test set there were 621 titles (incl. subtitles, roof titles and intermediary titles), and for the detection and labeling task the manual ruleset achieved a precision of 86% and a recall of 96% (resulting in an F-measure of 90.7%). For the detection of captions on the test set containing 255 instances, the rule set was able to achieve an F-measure of 97.6% (precision 98.4% and recall 96.8%). These values are only significant to show that a relatively simple rule set is able to perform quite well on known layouts, thus giving hope that such rule sets can be evolved in the future automatically through machine learning methods. Based on the results produced by these two manual rule sets, the article segmentation algorithm was able to correctly segment 85.2% of the 311 total articles present in the test set. While the vast majority of document images are segmented correctly, a few pages fail catastrophically, thus generating most of the observed errors (e.g. two pages were responsible for more than 75% of the title labeling errors). Article split errors were the most common, totaling 13.2% and most often these were generated as a direct consequence of a wrong page segmentation (i.e. split non-text regions, such as tables).

3 Conclusion (all)

-described 1. new theoretical model 2. working system 3. showed how to adapt such a system to a ML approach -future work: benchmark& data sets for logical layout analysis evaluation evaluation on different layouts - e.g. magazines, newspapers, books take into account publication as a whole (not just individual pages in order to account for multi-page articles/chapters)

References

O. Akindele and A. Belaid. Page segmentation by segment tracing. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, pages 341–344, 1993.

A. Antonacopoulos, B. Gatos, and D. Bridson. Icdar2007 page segmentation competition. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, volume 2, pages 1279–1283. IEEE Computer Society, 2007.

P. Awasthi, A. Gagrani, and B. Ravindran. Image modeling using tree structured conditional random fields. In *IJCAI (2007)*, 2007.

H.S. Baird and M.R. Casey. Towards versatile document analysis systems. In *Proc. 7th International Workshop Document Analysis Systems*, pages 280–290, 2006.

J. Besag. Statistical analysis of non-lattice data. the statistician, 24(3):179–195. *The Statistician*, 24(3):179–195, 1975.

T. M. Breuel. High performance document layout analysis. In *Symp. Document Image Understanding Technology, Greenbelt, Maryland*, page accepted for publication, 2003.

R. Cattoni, T. Coianiz, S. Messelodi, and C. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical Report 9703-09, ITC-irst, 1998.

S. Chen, S. Mao, and G.R. Thoma. Simultaneous layout style and logical entity recognition in a heterogeneous collection of documents. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, volume 1, pages 118–122. IEEE Computer Society, 2007.

Boris Chidlovskii and Loïc Lecerf. Stacked dependency networks for layout document structuring. In *SAC 2008*, pages 424–428, 2008.

A. P. Dias. Minimum spanning trees for text segmentation. In *Proc. Annual Symposium Document Analysis and Information Retrieval*, 1996.

D. Doermann. Page decomposition and related research. In *Proc. Symp. Document Image Understanding Technology*, pages 39–55, 1995.

F. Esposito, D. Malerba, G. Semeraro, S. Ferilli, O. Altamura, T.M.A. Basile, M. Berardi, M. Ceci, and N. Di Mauro. Machine learning methods for automatically processing historical documents: from paper acquisition to xml transformation. In *Proc. 1st International Workshop Document Image Analysis for Libraries*, pages 328–335. IEEE Computer Society, 2004.

J. Finkel, T. Grenager, and C.D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005*, 2005.

R. Furmaniak. Unsupervised newspaper segmentation using language context. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, volume 2, pages 619–623. IEEE Computer Society, 2007.

B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis. Automatic table detection in document images. In *Proc. 3rd Int. Conf. on Advances in Pattern Recognition (ICAPR'05), LNCS 3686*, pages 609–618, 2005.

Lise Getoor and Ben Taskar, editors. *Introduction to Relational Statistical Learning*. MIT Press, 2007.

J. Ha, R. Haralick, and I. Phillips. Document page decomposition by the bounding-box projection technique. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, pages 1119–1122, 1995.

R. Haralick. Document image understanding: Geometric and logical layout. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 385–390, 1994.

D. J. Ittner and H. S. Baird. Language-free layout analysis. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, pages 336–340, 1993.

A.K. Jain and B. Yu. Document representation and its application to page decomposition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):294–308, 1998.

K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on ine Learning, 2001*, 2001.

Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing, , Cambridge, MA, 1999*. MIT Press, Cambridge, MA, 1999.

S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms: A literature survey. In *Document Recognition and Retrieval X*, volume 5010, pages 197–207. SPIE, 2003.

S. Marinai and H. Fujisawa, editors. *Machine Learning in Document Analysis and Recognition*. Springer, 2008.

Jennifer Neville and David Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692, 2007.

D. Niyogi and S.N. Srihari. Knowledge-based derivation of document logical structure. In *Proc. Int. Conference on Document Analysis and Recognition*, pages 472–475, Montreal, Canada, 1995.

Gerhard Paaß and Frank Reichartz. Exploiting semantic constraints for estimating supersenses with crfs. In *Proc. SDM 2009*, 2009.

Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL 2004*, pages 329–336, 2004.

Y. Rangoni and A. Belaïd. Document logical structure analysis based on perceptive cycles. In *Proc. 7th International Workshop Document Analysis Systems*, pages 117–128. Springer, 2006.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

K. Pramod Sankar, Vamshi Ambati, Lakshmi Pratha, and C.V. Jawahar1. Digitizing a million books: Challenges for document analysis. In *Proc. 7th International Workshop Document Analysis Systems*, pages 425–436, 2006.

Karl-Michael Schneider. Information extraction from calls for papers with conditional random fields and layout features. *Artif. Intell. Rev.*, 25:67–77, 2006.

Faisal Shafait, Daniel Keysers, and Thomas Breuel. Performance comparison of six algorithms for page segmentation. In *7th IAPR Workshop on Document Analysis Systems (DAS)*, pages 368–379, 2006.

K. Summers. Near-wordless document structure classification. In *Proc. International Conf. on Document Analysis and Recognition (ICDAR)*, pages 462–465, 1995.

C. Sutton and A. McCallum. Collective segmentation and labeling of distant entities in information extraction. In *ICML workshop on Statistical Relational Learning (2004)*, 2004.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Relational Statistical Learning*, chapter An Introduction to Conditional Random Fields for Relational Learning. MIT Press, 2007.

Charles A. Sutton, Khashayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proc. ICML 2004*, 2004.

Y. Tang, H. Ma, X. Mao, D. Liu, and C. Suen. A new approach to document analysis based on modified fractal signature. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, pages 567–570, 1995.

Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *In Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

S. Tsujimoto and H. Asada. Major components of a complete text reading system. *Proc. IEEE*, 80(7):1133–1149, 1992.

J. van Beusekom, D. Keysers, F. Shafait, , and T.M. Breuel. Example-based logical labeling of document title page images. In *Proc. International Conf. Document Analysis and Recognition (ICDAR)*, volume 2, pages 919–923. IEEE Computer Society, 2007.

Luc Vincent. Google book search: Document understanding on a massive scale. In *Proc. 9th International Conf. Document Analysis and Recognition*, pages 819–823, 2007.

Paul A. Viola and Mukund Narasimhan. Learning to extract information from semi-structured text using a discriminative context free grammar. In *SIGIR 2005*, pages 330–337, 2005.

F. Wahl, K. Wong, and R. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Vision, Graphics, and Image Processing*, 20:375–390, 1982.

Guillaume Wisniewski and Patrick Gallinari. Relaxation labeling for selecting and exploiting efficiently non-local dependencies in sequence labeling. In *ECML PKDD 2007*, pages 312–323, 2007.