Computers and Electrical Engineering 40 (2014) 1363–1375

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Contextual modeling for logical labeling of PDF documents ☆

X. Tao [a], Z. Tang [a,b,*], C. Xu [b]

[a] Institute of Computer Science and Technology, Peking University, Beijing, China
[b] State Key Laboratory of Digital Publishing Technology, Beijing, China

ARTICLE INFO

Article history:
Available online 16 February 2014

ABSTRACT

The widely-used Portable Document Format (PDF) documents are known to be layout-oriented and not suitable for mobile applications. In this paper, a Conditional Random Fields (CRF) based model is proposed to learn latent semantics of PDF page content. Local and contextual observations constructed from PDF attributes are incorporated to facilitate the determination of semantic roles. The observations are carefully designed to work even in different styles of documents. A local classifier is first used to generate posterior probabilities. The local estimate is then fed to the CRF model for joint classification. The experimental results evidently approve the positive effects of contextual information in logical labeling. Our work has revealed the potential usability of existing born-digital fixed-layout documents for mobile applications.

1. Introduction

1.1. Background

With the prosperity of mobile devices like smartphones and tablets, it is preferable to accomplish daily necessities in a more portable way. Regarding the application scenario of mobile reading, documents in reflowable formats have become highly desirable. A reflowable document format allows the user to tune the content sizes to his satisfaction, and rearranges a moderate amount of content into the display region. The advantages of such document lie in the visual adjustability on mobile devices with different display sizes. Fixed-layout document format, on the other hand, specifies the position of each content precisely so that consistent layout is retained on various devices. However, the user has to zoom and drag in order to see the text clearly on smaller devices. Such reading experience hinders the usability of fixed-layout documents in mobile applications. The two formats are compared visually in Fig. 1.

In fact, a large amount of existing electronic documents are born-digital fixed-layout documents, typically in PDF format. A PDF document page can contain any possible combination of low level page elements such as text, graphics and images, whose layout and formatting is fully specified [1]. Considering the accurate content coordinates and precise text codes provided by the legacy PDF documents, they are well-suited materials for producing high-quality reflowable documents. Unfortunately, there are no explicit semantics defined in PDF documents.

Legitimate conversion is indispensable to enable the usability of born-digital fixed-layout documents for mobile reading. The conversion commonly involves segmenting the documents into smaller pieces and reorganizing them according to their semantic roles. The semantic roles have to be recognized properly and forwarded to the reflowable documents, so that the

☆ Reviews processed and recommended for publication to Editor-in-Chief by Guest Editor Dr. Jing Tian.
* Corresponding author at: 128 Zhongguancun Street, Haidian District, Beijing 100080, China. Tel.: +86 10 82529725.
E-mail addresses: jolly.tao@pku.edu.cn (X. Tao), tangzhi@pku.edu.cn (Z. Tang), ccxu09@yeah.net (C. Xu).

http://dx.doi.org/10.1016/j.compeleceng.2014.01.005
0045-7906/

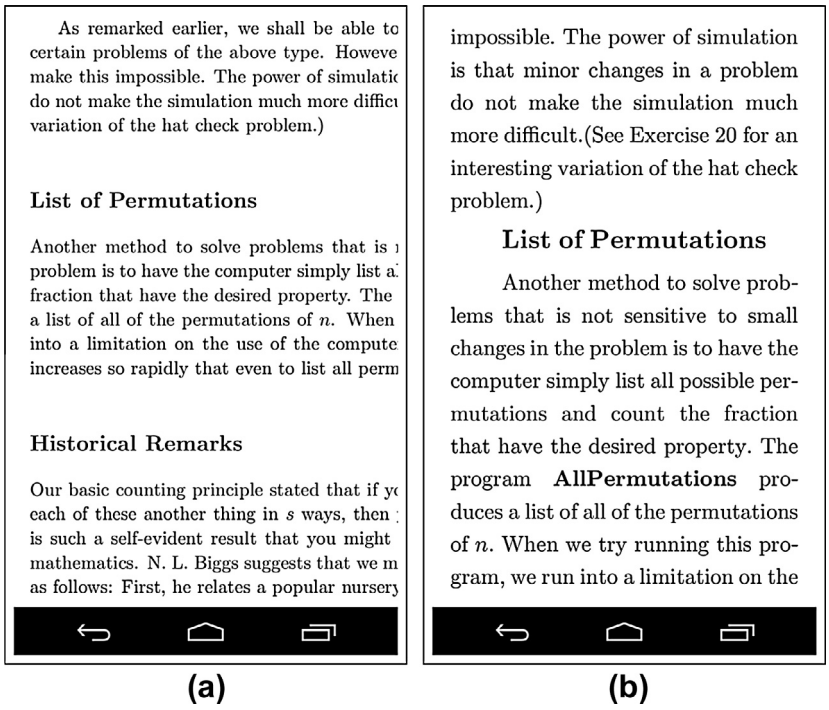


Fig. 1. (a) Screenshot of a fixed-layout document on a mobile device. Note that the page is enlarged to be visually comfortable. However the text on the right is cut out, because the original layout cannot fit into a smaller display size. In order to make them visible, the page has to be slid. (b) Reflowable document with the same content. The text is rearranged and correctly wrapped to adapt to the screen and the semantic roles of the content remain distinguishable.

content can be kept comprehensible. On the contrary, incorrect semantic roles will make the documents rendered in an unreasonable way, obscuring their real intentions. However, the crucial determination of the semantic roles of the content, also known as logical labeling, remains an open problem.

In this work, a 2D CRF framework is applied to model the hidden semantics of document page fragments. Raw observations are developed from attributes directly available from PDF documents. These observations are further extended to unary and pairwise features that represent local descriptions and contextual relationships. The rest of this section introduces related work and motivation of this work. The CRF framework is briefly explained in Section 2. The problem formulation and feature engineering is proposed in Section 3. Its application on a ground-truthed PDF document dataset and performance evaluation is presented in Section 4. The conclusion and future work are given in the last section.

1.2. Related work

Usability of fixed-layout documents in mobile application has drawn sufficient attention of researchers. Breuel et al. [2] presented a method that breaks document images down to word components and re-organizes them as HTML consisting of a sequence of image references so as to be reflowable on mobile devices. Method proposed by Erol et al. [3] extracts salient content from source documents, selects an optimal subset of elements, and composed them into a multimedia thumbnail playable through both visual and audio channels. Recently, markup language based document formats like Electronic Publication (EPUB) [4] prevail in the field of mobile reading. Marinai described a rule based system to identify the table of contents [5] and the notes in the text for converting certain text for converting certain books into EPUB format [6].

Plenty of research has been devoted to achieving reliable segmentation of document pages in the past decades. The segmentation algorithms may adopt top-down, bottom-up or hybrid strategies. Comprehensive surveys can be found in [7], and performance of these algorithms is evaluated in [8]. Compared with segmentation of fixed-layout documents, logical labeling has far less available literature due to its inherent complexity. DIVA research group proposed a reverse engineering tool [9] to analyze the embedded resources of PDF files and generate their physical structures in an intermediate format [10]. Then another interactive system [11] was presented to recover the logical structures through neural network learning mechanism. Rangoni and Belaïd [12] used an transparent artificial neural network and resolve ambiguous results through a feedback mechanism. With the help of an OCR engine, Luong et al. [13] uses a linear chain based CRF model to detect logical structures of documents from scholarly digital libraries. Tang's group focuses research on recognition of specific logical structures

involving paragraph [14], mathematical formula [15], and graphic component [16]. It is claimed that the logical labeling methods have no standardized benchmarks or evaluation sets [17], which is highly desired in this field.

1.3. Motivation

The task of logical labeling can be regarded as exploring the latent semantics of document page content objects. E-book pages, for instance, have a set of distinguishable logical classes such as titles, body text, and figures and tables.

Intuitively, it is possible to infer the semantic roles of some document content objects independently regardless of the rest of page. For instance, non-textual content often plays the part of illustration, and a number with few digits near page corner tends to be a page number. It is also noticed that a typesetting scheme is used to differentiate intra-page semantics of content objects. However, documents are generally designed in their own styles, and a single universal scheme is unlikely to exist.

When there is more variation in style, semantics are clarified more likely by relative relationships among neighbors. Two types of relationships are conceivable: the interactions between latent semantics reflect which logical classes conventionally appear together, even prior to the specification of their appearance; adjacent fragments exhibit relative similarities and diversities in visual perception for readability purpose. Modern typesetting systems generally use similar appearance styles for homogeneous content fragments, and make distinctions among heterogeneous ones. For example, text lines within same paragraphs give the impression of consistent text properties and line spacing, while titles are often designed specially to be distinguished from their neighbors.

Following the above facts, relational dependencies can expected to be exploited and learned by statistic models. Graphical model, as a probabilistic framework modeling dependencies between random variables, can naturally fit the aforementioned intuitions. Conditional Random Fields [18], a special form of graphical model, directly models the conditional probability distribution of labels given the observed data. CRF has already achieved extensive successes in various application fields like natural language processing [19], computer vision [20] and image document analysis [21]. Provided with adequate observations, we expect this framework is effective for logical labeling task of born-digital fixed-layout documents.

2. CRF framework

To assign a correct label for each physical fragment in a page, we can formulate this task as a classification problem. Let the fragments be indexed by i, Y_i be the multinomial random variable indicating the logical role of a fragment whose value can be taken from a label set $\mathcal{L}$, and X_i be the observations characterizing the fragment. The model $P(\boldsymbol{Y}|\boldsymbol{X})$ then describes the distribution of logical labels $\boldsymbol{Y} = \{Y_i\}$ given observations $\boldsymbol{X} = \{X_i\}$. A graph $G = \langle V, E\rangle$ can be built with each vertex associated with a random variable Y_i. $(\boldsymbol{X}, \boldsymbol{Y})$ is a Conditional Random Field if the variables $\boldsymbol{Y}$, when conditioned on $\boldsymbol{X}$, satisfy the Markov property with respect to G:

$$P(Y_i|\boldsymbol{X}, \boldsymbol{Y}_{V\setminus i}) = P(Y_i|\boldsymbol{X}, \boldsymbol{Y}_{N_i}), \tag{1}$$

where $V \setminus i$ denotes all vertices except i, and $N_i = \{j|(i,j) \in E\}$ is i's neighborhood. By the Hammersley and Clifford theorem, the conditional probability distribution $P(\boldsymbol{Y}|\boldsymbol{X})$ factorizes over G into unnormalized potential functions $\Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c)$ on maximal cliques

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{c\in G}\Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c) \tag{2}$$

where $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}'}\prod_{c\in G}\Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c)$ is the partition function summing over all possible assignments of $\boldsymbol{Y}$.

Taking the log linear form, the potential function can be parameterized as

$$\Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c; \boldsymbol{\lambda}_c) = \exp\left\{\sum_k \lambda_{ck} f_{ck}(\boldsymbol{x}_c, \boldsymbol{y}_c)\right\} \tag{3}$$

where $\{f_{ck}(\boldsymbol{x}_c, \boldsymbol{y}_c)\}$ are feature functions of clique c indexed by k. In order to reduce model complexity, the cliques can be further grouped into a set of clusters $\mathcal{C}$, where $C_p \in \mathcal{C}$ is called a clique template. Cliques belonging to a clique template C_p share the same parameters $\boldsymbol{\lambda}_p$.

3. Modeling

In this section, we first introduce available attributes from typical PDF files and define the fragments on which logical labeling is carried out. As stated above, the hidden semantics reside in not only the local appearance, but also the relationships among content objects. In light of such intuition, we formulate logical labeling as a joint classification problem using the CRF framework. For feature engineering, we develop basic observations based on PDF attributes, and extend them to richer observations. We carefully design these observations so that they are effective across diverse document styles. The schematic workflow of logical labeling task is illustrated in Fig. 2.

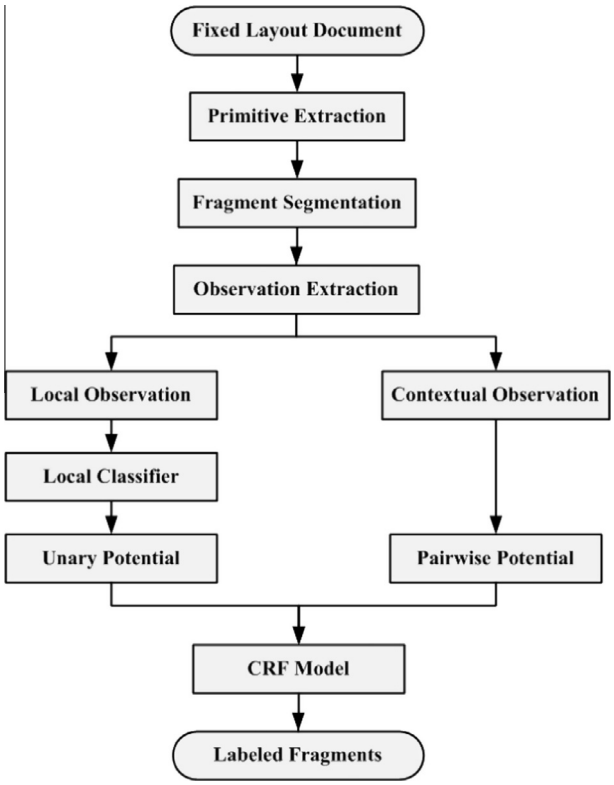


Fig. 2. Workflow.

3.1. PDF attributes and fragments

A PDF page is made up of primitive objects called content streams which are defined with rich attributes to determine and reproduce the page's appearance. These primitives describing visible content are categorized into text, images and graphics. Accurate geometric coordinates and bounding boxes of the three types of primitives are available to represent their positions and sizes in the page. Specifically, text primitives contain their character codes associated with state parameters like font families and font sizes. Image primitives are rectangular arrays of pixel values. Graphic primitives consist of instructions to render vector graphics. The paths of graphics are constructed with operations in form of straight lines, rectangles and cubic curves. Each operation contains an operator and necessary operands to form a precise definition. From another perspective, a page of fixed-layout document can be regarded as a holistic image represented by means of pixels. For each document page, both the embedded low level enriched content objects and sampled image of the entire page can be extracted by an eligible parser. In this work, we use a commercial PDF parser engine provided by Founder Corporation. Fig. 3 shows the low level attributes parsed from a sample page.

The primitives are the basic units to be clustered into fragments, which are defined as an aggregations of adjacent basic elements with size no larger than a text line. Fragments offer richer descriptions compared with content primitives and require less sophisticated physical segmentation algorithms than higher level blocks. Fragments are attainable as products of general physical segmentation by grouping the content primitives according to homogeneity and geometric adjacency. However, our work focus on the logical roles of the content and the fragments are manually segmented to exclude errors caused by physical analysis. Logical labeling is then carried out on the scale of fragments.

3.2. CRF formulation

We take both local and contextual evidence into consideration in order to obtain a discriminative model. The local observations and neighboring interactions are correspondingly expressed with unary and binary cliques in the CRF model. A label set $\mathcal{L}$ of total 16 semantic logical labels is defined, including body text, equation, figure, figure annotation, figure caption, figure caption continuation, list item, list item continuation, footer, header, marginal, notes, table cell, table caption, page number, title. Each fragment can be assigned with a corresponding logical label through model inference. Fig. 4(a) shows a sample page with ground-truth labels.

To treat a page as a graph, centroids of bounding boxes of fragments are extracted as vertices, and edges are created between vertices. With the edges measured by Euclidean distance, a minimum spanning tree (MST) is constructed to

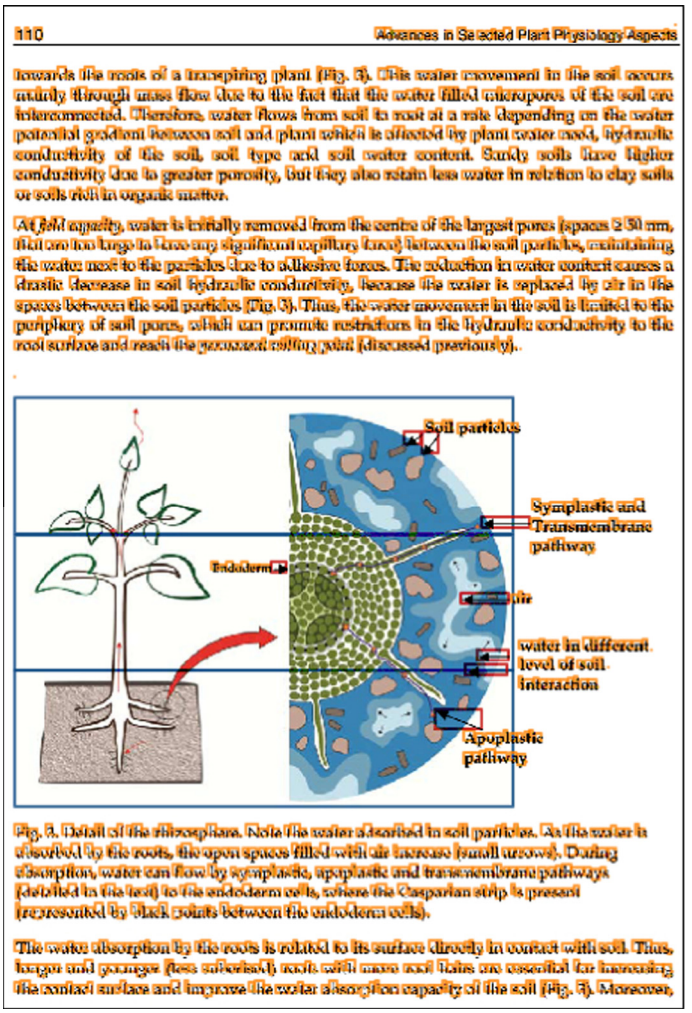


Fig. 3. Low level content objects available from PDF documents. Each text glyph is bounded with an orange box; the illustration is composed of three vertically adjacent images in blue boxes; the annotation arrows on the illustration are marked with red boxes as graphics. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

establish neighborhood of each fragment within the page. An example is illustrated in Fig. 4(b). The minimum spanning tree is a global optimum that ensures the sum of the edges distances is minimal among all possible spanning trees of the same graph. Each vertex i is attached with a random variable representing its logical label denoted as $y_i \in \mathcal{L}$. The observable data of i is denoted as $\boldsymbol{x}_i$.

Let $\{g_l(\boldsymbol{x}_i)\}$ define the local observations over fragment i. Then the unary feature functions can be defined as

$$f_{s,l}(y_i, \boldsymbol{x}_i) = \mathbb{1}\{y_i = s\} g_l(\boldsymbol{x}_i) \tag{4}$$

where $s, l \in \mathcal{L}$, and $\mathbb{1}\{y_i = s\}$ denotes an indicator function which equals 1 if $y_i = s$ and 0 otherwise. The potential function of unary cliques can be parameterized as

$$\Psi(y_i, \boldsymbol{x}_i) = \exp\left\{\sum_{s,l \in \mathcal{L}} \lambda_{s,l} f_{s,l}(y_i, \boldsymbol{x}_i)\right\} \tag{5}$$

where $\lambda_{s,l}$ is shared across all unary cliques.

In addition to local observations, we also exploit informative interactions to capture possible dependencies between fragments. Given observations $\{g_k(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$ of two connected random variables i, j in the graph, their pairwise feature functions are defined as

$$f_{s,t,k}(y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\{y_i = s, y_j = t\} g_k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6}$$

and the potential function is similarly parameterized as

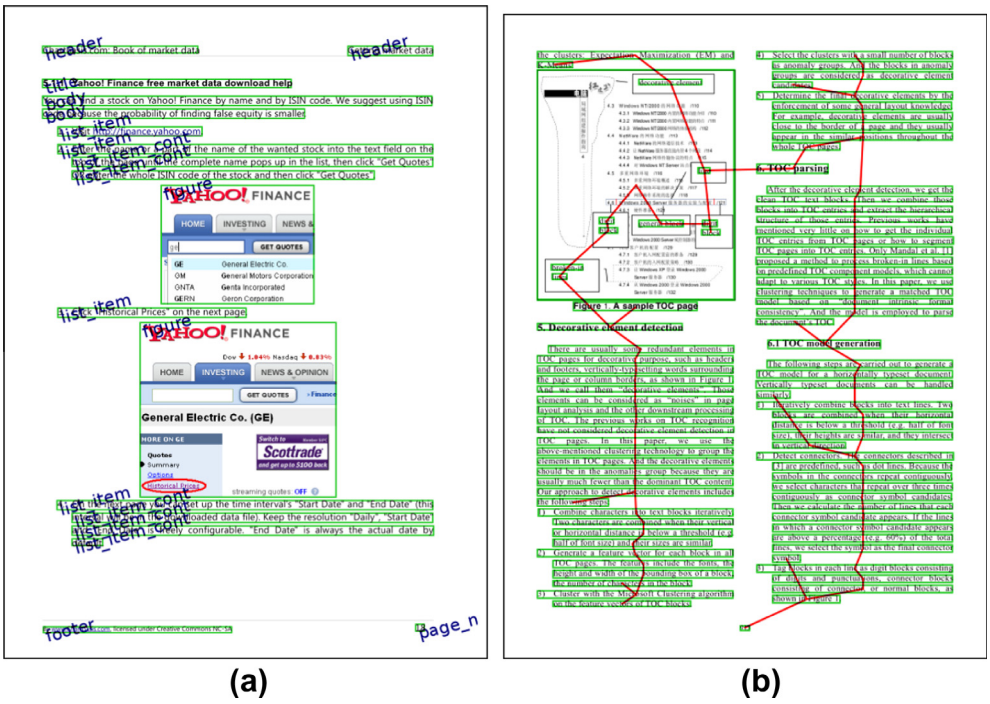


Fig. 4. (a) A ground-truth page with logical labels. Fragments are bounded with rectangle boxes. Logical labels are rendered in blue. (b) An example of minimum spanning tree graph constructed within a PDF document page. The graph structure is depicted by solid lines connecting the centroids of fragments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\Psi(y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{\sum_{s,t,k} \lambda_{s,t,k} f_{s,t,k}(y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)\right\} \tag{7}$$

3.3. Training and inference

Instead of exact likelihood, pseudolikelihood is used for parameter estimation for computational efficiency. For each random variable y_i, the labeling of its neighbors $\boldsymbol{y}_{N_i}$ is regarded as fixed ground-truth, and the distribution of y_i depends on factors containing y_i. The computation of local partition function only involves summing over the possible labels of y_i itself:

$$p_{PL}(y_i|\boldsymbol{y}_{N_i}, \boldsymbol{x}) = \frac{\prod_{i\in C}\Psi_c(\boldsymbol{y}_c, \boldsymbol{x}_c)}{\sum_{y'}\prod_{i\in C}\Psi_c(\boldsymbol{y}'_c, \boldsymbol{x}_c)}$$

Let N be the number of cliques in the graph, S be the number of possible states (logical labels) for each random variable. In this problem, the cliques have the order of at most 2, so the computational complexity of objective function for a page is $O(SN)$. The computation of gradient requires $O(SNF)$, where F is the number of feature functions, since gradient is computed with respect to model parameters associated with feature functions.

The training is performed by maximizing the log of conditional pseudolikelihood (PL) using ℓ_2 with respect to $\boldsymbol{\lambda}$:

$$\ell_{PL}(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}) = \sum_{i\in V} \log p_{PL}(y_i|\boldsymbol{y}_{N_i}, \boldsymbol{x}; \boldsymbol{\lambda}) - \tau\|\boldsymbol{\lambda}\|^2$$

Parameter τ penalizes weight vector $\boldsymbol{\lambda}$ with too large magnitude. The ℓ_2 norm also allows easier computation of gradient, making the regularization naturally fit into gradient based optimization. The maximization is accomplished by a quasi-Newton optimization method L-BFGS [22], which gradually adjusts the parameters iteratively until convergence.

We choose Belief Propagation (BP) as the inference algorithm for prediction. Belief Propagation is an inference algorithm using message passing scheme. It can be used to estimate marginals or most likely states. When there are no loops in graph, BP provides exact solutions. With messages passed in proper order, the number of message updates is linear to the number of cliques for acyclic graph. Inference using BP on our tree graph requires $O(N)$ message updates and each message costs $O(S^2)$, thus the total complexity is $O(S^2N)$. This is much more efficient than the brute force way with prohibitively expensive cost exponential to S. Labels of fragments are predicted as the assignments to $\boldsymbol{y}$ that maximize $p(\boldsymbol{y}|\boldsymbol{x})$.

3.4. Feature engineering

To provide the model with effective information, feature selection and engineering is of importance. This subsection expounds the preprocessing procedure, the design of raw observations, and their extension to more exquisite observations.

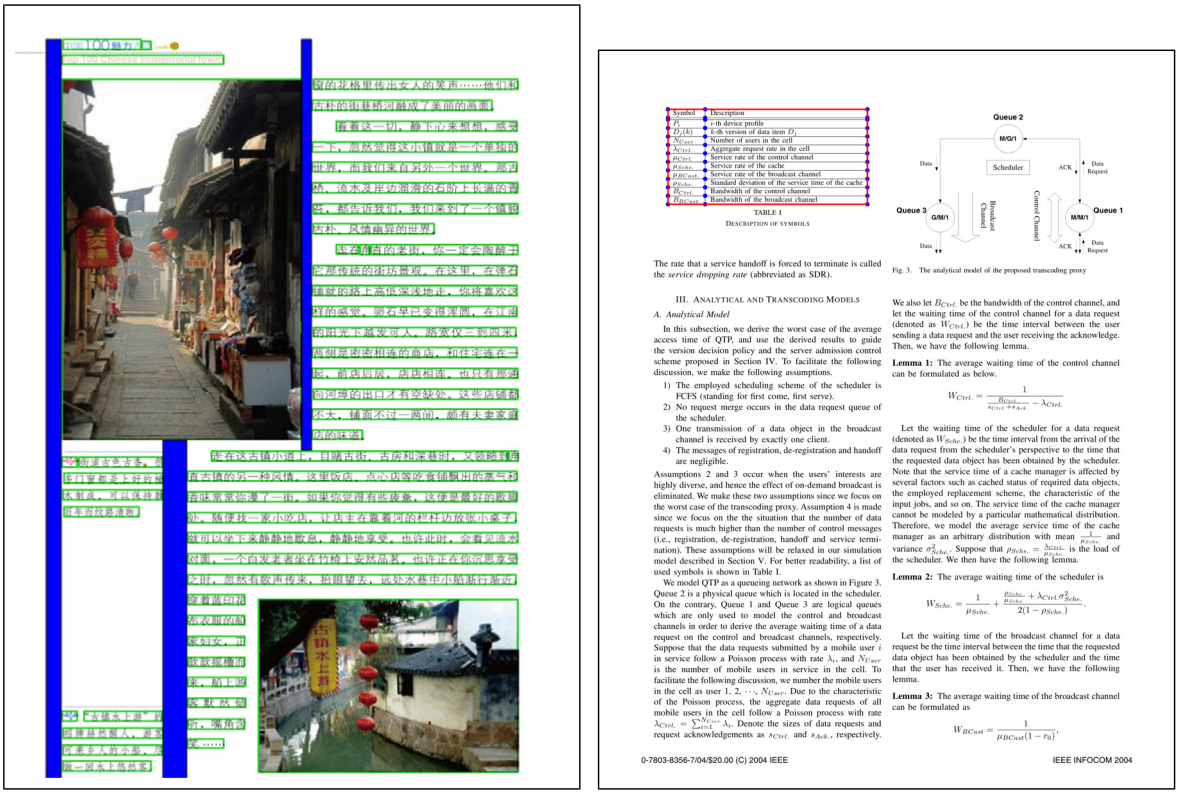
3.4.1. Preprocessing

Before developing the feature set, we first carry out peripheral analysis to gain auxiliary knowledge in page scale, covering dominant font size, column separators and regions of interest for specific logical class, etc.

Dominant font size. There is no fixed typesetting scheme for font size assignment. It is observed that relative font size distinctions occur between most different logical classes. Take an example, font size of title often appear to be larger, while that of header or footer is smaller. Rather than using absolute font size, a baseline for relative font size comparison is more appropriate. In our work, the most frequently used font size is determined as the baseline, namely dominant font size.

Column separator. The indent variation is a useful indication for possible change of semantics. To acquire fragment indent, column separators are indispensable, especially in cases of multi-columned pages. The separators delimit the left and right boundaries of columns. To detect the candidates of column separators, we adopt the whitespace covering method [23]. It iteratively divides the page into four sub-rectangles and find the ones without fragment inside as column separator candidates. These candidates are further filtered by their shapes so that only tall and thin whitespaces are kept. Fig. 5(a) shows an example of successful detection of three column separators in a page with non-Manhattan layout.

Table region candidate. Logical class like table cell cannot be easily recognized just depending on the observations over a single fragment and its neighbors. Rather, the whole table region gives a stronger impression of regular grids. Table regions show apparent horizontal and vertical ruling lines as visual separators. To capture table line grids, We use second order derivative filters to detect all vertical and horizontal lines along with their intersections on each page image, as is shown in Fig. 5(b). The number of the intersecting points of lines are considered to rule out false table lines. The assumption behind is that a table contains more than one cell. The bounded table border are employed to judge whether a text fragment is within a possible table bounding box, which is added as a feature to boost table cell recognition.



(a) Column separators (b) Table grids

Fig. 5. (a) An example of column separator detection in a page with non-Manhattan layout. Three blue candidates successfully delimit the left column bounds. (b) An example of table region detection based on the grids of table lines. The blue points are the intersections of table lines. The table region is detected as the red rectangle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4.2. Raw observations

For each fragment, a set of basic observations, which are intuitive and straightforward to calculate, are termed as raw observations, mainly including geometric, textual, typesetting and visual observations. The detail descriptions are given in Table 1.

Geometric. Geometric measures are the most direct descriptions for fragments. In fact, the semantic logical classes are visually different due to their positions and shapes. Majority of the fragments in a document page belong to body text, which are homogeneous in terms of shape and position. Another obvious example is that figure fragments usually have larger area sizes, while fragments like footnotes and marginals are much smaller. To reflect and describe these observations, we can extract locations and geometric shapes to provide information for the model learner.

- Rectangles. The heights, widths and areas of fragment rectangles are calculated and then normalized by medians, so that they are not restricted by content sizes of specific document styles. It is expected that these observations can contribute to differentiate labels such as body text, figure and footer.
- Aspect ratio. The aspect ratio of fragment bounding box is used to roughly describe the shape of a fragment.
- Position. It is noticed that functional fragments like page numbers, headers and footers are always laid at outer boundaries of the whole page. Relative position of a fragment within a page is profitable in this regard.

Textual. As rigorously formatted text documents, PDF can provide richer representation to text, which can be exploited in constructive ways. Some of the logical classes come with explicit text patterns. Taking advantage of precise text codes, we can describe classes more concretely.

- General patterns. General patterns here refer to characteristics of digits, capitalization and punctuation, etc. These observations help to indicate that a fragment is page number or title.
- Figure caption patterns. Figure captions generally start with patterns like "Figure" or "Fig.", followed by a number. There also exists figure captions without this pattern, which needs other expressions to be identified.
- List item patterns. List items start with symbolic or numerical bullets. The detection of bullets such as dots, dashes and alphanumeric points, is a determining factor for the identification of first line of list item. The consistency of bullets and list item continuation is expected to be learned in pairwise contextual representation.
- Equation patterns. Equations contain mathematical symbols and Greek letters.

Typesetting. Benefiting from rich attributes of fixed-layout documents, we pay special attention in the inherent attributes which contribute to performance.

- Font. Font information is available for each fragment. For each textual fragment, we compare its font size with the dominant font size (see Section 3.4.1), and categorize it relatively as greater than, equal to and less than the dominant font size.

Table 1
Raw observations from PDF attributes.

Feature type	Feature description
Geometric	Normalized fragment height
	Normalized fragment width
	Normalized fragment area
	Aspect ratio of width and height
	Relative position within page
Textual	Has digit
	All digit
	Is uppercase
	Has math symbols or Greek letters
	Digital number pattern
	Figure caption pattern
	List item pattern
	Table caption pattern
	Title pattern
	Ends with sentence terminal
Typesetting	Font size greater than the dominant font size
	Font size smaller than the dominant font size
	Font size equal to the dominant font size
	Binarized discretized indent level
	Is fragment fully filled on right side
	Raw content source type of fragment, e.g. text, image or path
Visual	Gabor energy features describing fragment image texture

- Indent. Indent levels of the fragments are quantized left indentations. For each fragment, the distance between the left column bound and fragment left border is calculated as relative indent. The relative indents are discretized by assigning them to equal-sized bins with width of half the dominant font size. The first four non-empty bins are kept as significant levels. Other greater indents fall into a fifth level, suggesting they are too large to be distinguishable. Hence, the indent observation is a binary vector of length five, with one effective indent level.

Visual. In this work, image based observations considered are mainly image texture for the reason that most logical labels are related to text which indicate that the differentiation of text and non-text is more significant, especially for figure recognition. Figures, also called graphic composites in PDF including intensity images, graphics embedded or surrounded with text elements. In most cases, figures are made from small image primitives and various path operations. Complex cases in figures consist of combination of path, image and text annotation are an unavoidable challenge for figure classification.

With the fragment bounding box coordinates at hand, we can crop each fragment image from the page image. Then, each cropped fragment image is filtered with a set of local 2D Gabor filters with symmetric convolution kernels. Gabor images can be obtained by applying different preferred orientations and scales covering appropriately spatial frequency domain. In our experiments, 4 orientations and 1 scales comprise 4 Gabor filters, which produce a 8-dimensional vector for each point of each fragment image. The output 8 Gabor images are further processed to calculate Gabor energies. We bin the Gabor energy of response of the linear symmetric Gabor filters into 3 intervals, which results in 12-dimensional texture vector in total.

3.4.3. Unary observations

Raw observations characterize the fragments' properties on their own. Underlying semantic of current fragment is reflected in the observations of its neighbors and their relationships. Relative similarities and diversities in the fragment local neighborhood, are believed to be profitable for information gain. It is also reported that by considering local spatial and temporal neighborhood information, an increase of performance can be obtained on text and non-text classification of handwriting documents [24]. As is shown in Table 2, we integrate a set of 29 complementary local context observations derived from four types of raw observations.

Making use of pragmatic local classifiers, we can establish initial estimate based on observations over individual content fragments. A complete set of 73 observations, integrating raw observations and local context observations, are fed to local classifiers as inputs. We define the g_l function in Eq. (4) as

$$g_l(y_i, \boldsymbol{x}_i) = \log(p_{local}(y_i = l|\boldsymbol{x}_i)) \tag{8}$$

where $\boldsymbol{x}_i$ include both raw observations of i and its local context. In this way, the unary potentials of CRF model can be constructed as if the estimates of local classifiers are already observable. In this work, Support Vector Machine and Random Forest are used as local classifiers. Performance of these classifiers is compared in Section 4.2.

3.4.4. Pairwise observations

Observations between two fragments are the base of pairwise observation functions for CRF model. To capture the interactions between latent semantics, an observation g_0 is defined as constant value 1. Combining g_0 with an indicator function, the pairwise feature function $f_{s,t,0}$ in Eq. (6) is used to represent the co-occurrence of logical label pair (s, t). The value of associated parameter $\lambda_{s,t,0}$ accords with the frequency of (s, t) in the training set. The rest observations describe the relationships

Table 2
Observations from local context.

Feature type	Feature description
Geometric	Relative line spacing Eucilidean distance
Textual	Above fragment starts with figure caption pattern Above fragment starts with table caption pattern Above or below fragment starts with list bullet pattern Above fragment ends with sentence terminal Above ends with sentence terminal
Typesetting	Has above or below fragment Font size of above or below fragment greater than the dominant font size Font size of above or below fragment smaller than the dominant font size Font size of above or below fragment equal to the dominant font size Above or below fragment has same indent level Above or below fragment has deeper indent level Above or below fragment has shallower indent level Weighted relative indent hist of spatial neighborhood Above or below fragment fully filled on right side Above fragment belongs to raw content source image Fragment within raw content source image or path
Visual	Is within the image-based detected table bounding box

Table 3
Pairwise observations.

Geometric	Overlap between fragments One fragment contain the other Height ratio between fragments Width ratio between fragments Area ratio between fragments
Typesetting	Relative line spacing Eucilidean distance Alignment properties including left, right or central alignment Have same font id Equal font size between fragment
Visual	Gabor texture distance between fragments

of appearance between two adjacent fragments, by measuring the similarities in geometry, typesetting, and visual perception. All the pairwise observations are shown in Table 3.

4. Experimental results and discussion

4.1. Experimental setup

Our experimental data consists of a collection of 244 PDF document pages selected from 35 e-books in English and Chinese, which cover a wide range of layout styles for assessing the learning ability of the proposed model. Chinese books come from Founder Apabi digital library, and English books are selected among books crawled from web. The types of these books range from social or scientific library books to academic journals and magazines. It is known that there exists no standardized benchmarks or evaluation sets, which is time consuming to construct. However, we provide a ground-truthing tool to fill this gap in labeling process. A GUI application based on wxpython is developed to facilitate manual annotation of the dataset, which is accessible publicly from http://www.icst.pku.edu.cn/cpdp/data/marmot_data.htm.

Total 11347 fragments are manually marked by using the ground-truthing tool, and the physical fragments are further tagged with a set of 16 semantic logic labels, including `body text`, `title`, `figure`, `figure annotation`, `figure caption`, `figure caption continuation`, `list item`, `list item continuation`, `table cell`, `table caption`, `equation`, `page number`, `footer`, `header`, `footnote`, and `marginal note`. The total 244 PDF document pages are divided randomly into training and testing sets in a ratio of 2:1.

The performance is evaluated on the fragments using precision P, recall R and F_1-measure defined as $\frac{2 \cdot P \cdot R}{P+R}$. Among 16 semantic labels, as can been seen from Table 5, the distribution of fragments over each semantic label is highly imbalanced. Majority of the fragments belong to body text, which in this experimental setting possess a percentage of 50.7%. Hence, accuracy measure results can be misleading. More comprehensive metrics including macro- and micro-averaged F_1 are used respectively. Macro-averages weigh each label equally and compute their arithmetic mean, and micro-averages weigh each fragment equally and calculate the arithmetic mean.

4.2. Local classifiers

The local classifiers we adopt in this work are Support Vector Machine (SVM) and Random Forest (RF). These classifiers are trained with raw observations and additional local contextual observations. For (SVM) classifier, we choose linear and Radical Basis Function (RBF) kernels. The penalty parameter C and kernel coefficient γ are determined through grid searching on training set using a 5-fold cross validation scheme. RF classifiers are built with various numbers of trees in the forest. Increase in number of trees results in better performance but also longer training time, so a number is chosen where the performance stops improving significantly.

The three types of classifiers are trained using only raw observations as a baseline. Then additional observations of local context are appended to train the models that provide local estimates for CRF model. The performance of local classifiers and their hyperparameters selected through validation are shown in Table 4. Among all three types of local classifiers, RF achieves the best performance in both cases of raw and contextual observations. Also note that for all types of classifiers, addition of local context observations improves their performance (about 3–5% in micro-averaged F_1 and 5–10% in macro-averaged F_1).

In order to construct local observation functions as formulated in Eq. (8), the outputs of SVM classifier are converted into posterior probabilities using Platt's [25] method. Probability estimates of RF on an input sample are computed as mean probabilities predicted by trees in the forest.

4.3. Overall performance

In order to find out the optimal combination, all local classifiers trained with extended observations are validated. The final model achieving the best performance is a CRF model using RF as local estimator with regularization parameter

Table 4
Performance of local classifiers on testing set.

Observation	Local classifier	Micro-averaged F_1	Macro-averaged F_1
RAW	SVM(linear, $C = 10$)	85.60	71.13
	SVM(RBF, $C = 1000$, γ=0.001)	84.33	70.33
	RF(#tree = 1024)	87.72	77.60
CONTEXTUAL	SVM(linear, $C = 1$)	90.42	80.13
	SVM(RBF, C=1000, $\gamma = 0.001$)	91.79	82.70
	RF(#tree = 1024)	93.30	86.10

$\tau = 1$. Table 5 summarizes the performance of three models with different configuration. The first model RF-RAW is an RF classifier using only raw observations as input. In the second model RF-LOCAL, local context observations are appended to the feature set. The third model CRF-RF is a second order CRF model incorporating both probability estimates of RF-LOCAL and pairwise potentials.

Compared with the RF-RAW, RF-LOCAL has an increase of 5.58% in micro-averaged F_1 and 8.50% in macro-averaged F_1. With the aid of pairwise observations, the CRF model improves these measures further by 0.41% and 1.14% respectively. CRF model achieves the best overall performance on both micro- and macro-averaged F_1 measures. The effects of contextual modeling are empirically analyzed in the next subsection.

4.4. Effects of contextual information

Contrast between the two RF models reveals the advantages of local context observations. Comparing RF-LOCAL with RF-RAW in Table 5, and `list item continuation` gets the most impressive performance improvement. Given only raw observations, it is difficult to tell `list item continuation` apart from `body text` indeed. We believe that observations of local context like indent levels and textual patterns from neighbors (list item bullets, for example) contribute to this improvement. `Table cell` also achieve considerable improvement, benefiting from the fact that a table cell is usually inside a grid consisting of table lines, which is detected in the preprocessing stage. Similar justification applies to the situation of `figure annotation`. Characteristics related to nearest neighbors of `footer`, `title` and `marginal` account for their improvement. For example, a footer does not have a neighbor below, and it has smaller font size than its neighbor above.

Investigating the performance of CRF-RF model, performance on most labels demonstrate further improvement, compared with the results of RF-LOCAL. Performance on labels like `figure caption`, `list item` and `title` is improved in various degrees. It is inferred that common combinations are learned by CRF model to remedy local estimates. For example, a `figure caption` usually appears below a `figure`. The results approve the CRF model's ability to capture frequent patterns.

In Fig. 6, we illustrate some sample results of models including RF-RAW, RF-LOCAL and our proposed CRF-RF. In this sample page, most fragments belong to list items and list item continuations, which have two hierarchical levels in depth.

Table 5
Comparative performance of RF-RAW, RF-LOCAL and CRF-RF methods. In each row, the bold values denote the best F_1 measures achieved by RF-RAW, RF-LOCAL and CRF-RF.

Label	#Frag	RF-RAW			RF-LOCAL			CRF-RF		
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Body	5752	89.25	98.30	93.56	93.65	98.70	96.11	95.59	97.60	**96.59**
Equation	438	78.85	86.62	82.55	83.22	87.32	85.22	82.78	88.03	**85.32**
Figure	265	100.0	98.65	**99.32**	100.00	96.10	98.01	98.67	96.10	97.37
FigureAnnot	474	74.09	59.58	66.05	93.43	78.06	85.06	89.73	84.81	**87.20**
FigureCap	243	82.81	72.60	77.37	95.16	80.82	87.41	95.31	83.56	**89.05**
FigureCapCont	223	56.52	24.07	33.77	80.00	44.44	**57.14**	60.42	53.70	56.86
Footer	39	68.75	84.62	75.86	91.67	84.62	88.00	85.71	92.31	**88.89**
Header	262	92.39	97.70	94.97	93.41	97.70	95.51	91.58	100.00	**95.60**
ListItem	198	91.49	54.43	68.25	98.08	64.56	77.86	89.71	77.22	**82.99**
ListItemCont	320	64.41	40.00	49.35	87.50	88.42	**87.96**	80.37	90.53	85.15
Marginal	121	100.0	73.17	84.51	97.62	100.00	**98.80**	93.18	100.00	96.47
Note	69	100.0	88.46	**93.88**	100.00	69.23	81.82	95.65	84.62	89.80
PageNum	235	94.81	98.65	96.69	93.59	98.65	96.05	93.67	100.00	**96.73**
TableCap	64	88.89	44.44	59.26	100.00	44.44	61.54	90.00	50.00	**64.29**
TableCell	2395	88.87	85.25	87.02	93.59	95.02	94.30	95.65	93.90	**94.76**
Title	249	82.98	75.73	79.19	95.35	79.61	86.77	93.55	84.47	**88.78**
Micro-Averages	–	87.72	87.72	87.72	93.30	93.30	93.30	93.71	93.71	**93.71**
Macro-Averages	–	84.63	73.89	77.60	93.52	81.73	86.10	89.47	86.05	**87.24**

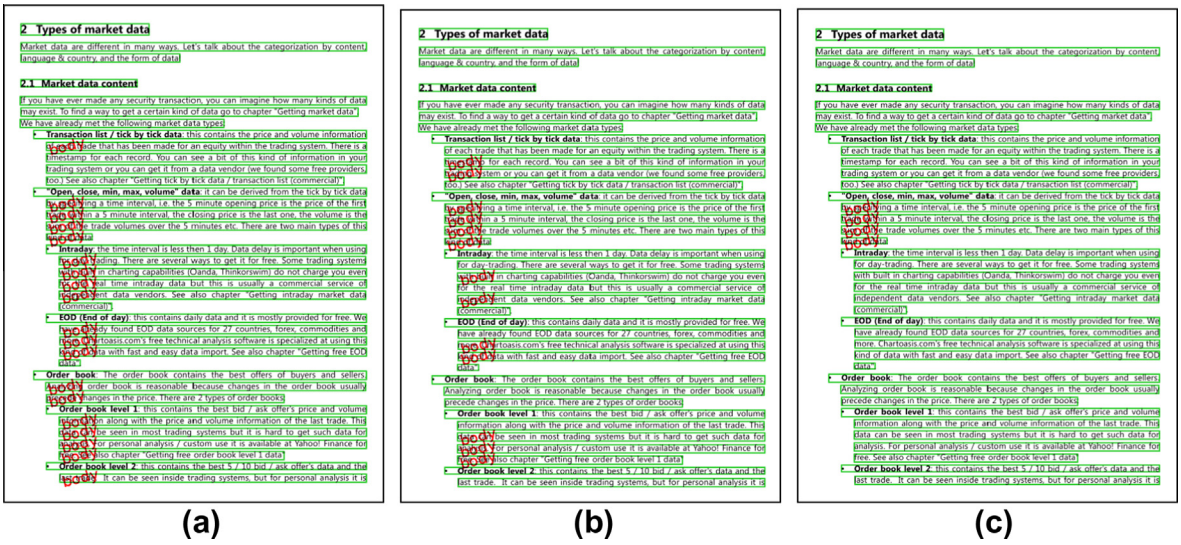


Fig. 6. Sample results for RF-RAW, RF-LOCAL and CRF-RF models. Most of the list item continuations are wrongly labeled by RF-RAW in (a). Some of them are corrected by RF-LOCAL in (b) due to local context. After learning the cooccurrence of neighboring list item and list item continuation, CRF-RF makes further improvements in (c).

The appearance of list item fragments is easily confused with body text fragments, which is also the result of basic RF-RAW model in Fig. 6(a). The local contextual information of neighboring fragments and the relative indent histogram of spatial neighborhood benefit the correction of misclassified fragments in Fig. 6(b). Fig. 6(c) shows that CRF-RF model pushes further improvement by adding complementary contextual relationships.

`Figure` and `note` are quite distinguishable in their own right, so contextual information does not help to improve their performance. Though performance of `figure caption continuation` and table caption also get improved in RF-LOCAL model, it is still far from satisfactory. This is ascribed to lack of pertinent modeling for their features and context.

5. Conclusion and future work

In this paper, we addressed the problem of logical labeling of born-digital fixed-layout documents. The proposed CRF model manages to identify the logical classes of document fragments by exploiting not only their local observations but also contextual information. All the observations are obtainable based on inherent PDF attributes and carefully designed to avoid being limited to a certain document style. The experimental results verify our hypothesis that the latent semantics of document fragments are better recognized by the aid of their context. Our work has revealed a promising prospect toward converting existing born-digital fixed-layout documents to reflowable documents, which may proliferate mobile reading.

We will enhance our model in several ways. Elaboration of additional discriminative features is expected to improve the performance. The model could also be augmented with context types other than spatial adjacency to capture more relationships. Our work is a preliminary step to the recovery of the entire logical structure of a fixed-layout document, which is required to approach the goal of practical use of documents in reading and retrieval. For future work, it is worthy to challenge higher-level semantics spreading multiple pages.

Acknowledgment

This work was supported by National Basic Research Program of China (No. 2012CB724108).

References

[1] Incorporated AS. Pdf reference, 6th ed.: Adobe portable document format version 1.7; 2006. <http://www.adobe.com/devnet/pdf/pdf_reference.html>.
[2] Breuel TM, Janssen WC, Popat K, Baird HS. Paper to pda. Proceedings. 16th international conference on pattern recognition, 2002., vol. 1. IEEE; 2002. p. 476–9.
[3] Erol B, Berkner K, Joshi S. Multimedia clip generation from documents for browsing on mobile devices. IEEE Trans Multimedia 2008;10(5):711–23.
[4] IDPF2013. Epub international digital publishing forum; 2013. <http://idpf.org/epub>.
[5] Marinai S, Marino E, Soda G. Table of contents recognition for converting pdf documents in e-book formats. In: Proceedings of the 10th ACM symposium on document engineering; 2010. p. 73–6.
[6] Marinai S, Marino E, Soda G. Conversion of pdf books in epub format. In: International Conference on Document Analysis and Recognition (ICDAR); 2011. p. 478–82.
[7] Mao S, Rosenfeld A, Kanungo T. Document structure analysis algorithms: a literature survey. In: Electronic imaging 2003, international society for optics and photonics; 2003. p. 197–207.

[8] Shafait F, Keysers D, Breuel TM. Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 2008;30(6):941–54.
[9] Hadjar K, Rigamonti M, Lalanne D, Ingold R. Xed: a new tool for extracting hidden structures from electronic documents. In: Proceedings of international workshop on document image analysis for libraries; 2004. p. 212–24.
[10] Bloechle J, Rigamonti M, Hadjar K, Lalanne D, Ingold R. Xcdf: a canonical and structured document format. In: Document analysis systems VII. Springer; 2006. p. 141–52.
[11] Bloechle J, Rigamonti M, Ingold R. Ocd dolores-recovering logical structures for dummies. In: 10th IAPR International Workshop on Document Analysis Systems (DAS); 2012. p. 245–249.
[12] Rangoni YY, Belaïd A. Document logical structure analysis based on perceptive cycles. In: Document analysis systems VII; 2006. p. 117–128.
[13] Luong M, Nguyen T, Kan M. Logical structure recovery in scholarly articles with rich document features. Int J Digital Lib Syst (IJDLS) 2010;1:1–23.
[14] Fang J, Tang Z, Gao L. Reflowing-driven paragraph recognition for electronic books in pdf. In: IS&T/SPIE electronic imaging; 2011. p. 78740U–78740U.
[15] Lin X, Gao L, Tang Z, Lin X, Hu X. Mathematical formula identification in pdf documents. In: International Conference on Document Analysis and Recognition (ICDAR); 2011. p. 1419–23.
[16] Xu C, Tang Z, Tao X, Li Y, Shi C. Graph-based layout analysis for pdf documents. In: IS&T/SPIE electronic imaging; 2013. p. 866407.
[17] Paaß G, Konya I. Machine learning for document structure recognition. In: Modeling, Learning, and Processing of Text Technological Data Structures; 2012. p. 221–47.
[18] Sutton C, McCallum A. An introduction to conditional random fields for relational learning. Introduction Stat Relational Learn 2007;93:142–6.
[19] Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth International Conference on Machine Learning, ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2001. p. 282–9.
[20] He X, Zemel RS, Carreira-Perpinán MA. Multiscale conditional random fields for image labeling. Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, vol. 2. IEEE; 2004. p. II–695.
[21] Nicolas S, Dardenne J, Paquet T, Heutte L. Document image segmentation using a 2d conditional random field model. Ninth international conference on document analysis and recognition, vol. 1. IEEE; 2007. p. 407–11.
[22] Liu D, Nocedal J. On the limited memory bfgs method for large scale optimization. Math Program 1989;45(1–3):503–28.
[23] Breuel TM. High performance document layout analysis. In: Proc symp document image understanding technology; 2003. p. 209–18.
[24] Delaye A, Liu C. Context modeling for text/non-text separation in freeform online handwritten documents. In: IS&T/SPIE electronic imaging; 2013. p. 86580C.
[25] Platt J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classifiers 1999;10(3):61–74.

Xin Tao received his B.Sc. degree in Computer Science in 2008 from Peking University, China. He is currently a Ph.D. student at Institute of Computer Science & Technology of Peking University. His research interests include document analysis and understanding.

Zhi Tang received his Ph.D. degree in Computer Science in 1995 from Peking University, China. He is now a professor at the Institute of Computer Science & Technology, Peking University and the director of the State Key Laboratory of Digital Publishing Technology. His research interests majorly include document analysis and understanding, digital rights management.

Canhui Xu received her Ph.D. degree in Computer Application from Central South University in 2011. Currently, she works in Qingdao University of Science and Technology. Her research interests include machine learning and image processing.