Вычислительные технологии, 2022, том 27, № 5, с. 69–78. © ФИЦ ИВТ, 2022
Computational Technologies, 2022, vol. 27, no. 5, pp. 69–78. © FRC ICT, 2022
ISSN 1560-7534
eISSN 2313-691X

DOI:10.25743/ICT.2022.27.5.007

Сегментация текста неразмеченных PDF-документов

А. О. Шигаров*, В. В. Парамонов
Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 664033, Иркутск, Россия
*Контактный автор: Шигаров Алексей Олегович, e-mail: shigarov@icc.ru
Поступила 08 июня 2022 г., принята в печать 30 июня 2022 г.

Большой объем нередактируемых документов публикуется и распространяется в формате PDF. Часто они являются “неразмеченными”, т. е. не сопровождаются аннотацией о собственной структуре, в них нет метаданных о месторасположении заголовков, параграфов, абзацев, таблиц, списков, рисунков, колонтитулов и пр. Анализ компоновки документов состоит в распознавании перечисленных элементов структуры. Базовой частью этого процесса является сегментация текста внутри страниц на блоки, которые затем можно классифицировать как заголовки, абзацы, ячейки таблиц и пр. Известные алгоритмы сегментации страниц в основном предназначены для работы либо с растровыми изображениями документов, либо с печатно-ориентированным ASCII-текстом. По сравнению с этими форматами данных PDF предоставляет дополнительную информацию (порядок рендеринга, шрифтовые метрики, линейки и пр.), которая может улучшить качество анализа компоновки документов. В работе излагается опыт адаптации некоторых существующих алгоритмов сегментации текста внутри страниц изображений документов и ASCII-текста, для того чтобы сделать их применимыми напрямую к формату PDF — неразмеченным случаям.

Ключевые слова: анализ компоновки документов, сегментация страниц, изображения документов, PDF-доступность, обработка документной информации.

Цитирование: Шигаров А.О., Парамонов В.В. Сегментация текста неразмеченных PDF-документов. Вычислительные технологии. 2022; 27(5):69–78. DOI:10.25743/ICT.2022.27.5.007.

Введение

PDF (Portable Document Format, `https://www.iso.org/standard/63534.html`) — один из наиболее популярных форматов нередактируемых документов. В 2015 г. его разработчики (P. Ydens) подсчитали, что количество PDF-файлов, ежегодно создаваемых в мире, равно примерно 2.5 триллионам (`https://www.pdfa.org/pdf-in-2016-broader-deeper-richer`). Многие из них являются неразмеченными, т. е. они не включают аннотацию собственной структуры (нет машиночитаемого представления заголовков, абзацев, таблиц, списков, надписей и пр.). Такой документ закодирован в виде низкоуровневых инструкций воспроизведения текста и графики на абстрактном графическом устройстве. Этого достаточно для того, чтобы PDF-документ мог быть прочитан человеком. Однако считывание его структуры компьютерной программой никак не обеспечивается.

J.T. Nganji [1] приводит следующую оценку: 95.5 % научных статей, публикуемых ведущими издателями, являются неразмеченными PDF-документами. Широкое распространение делает их ценным источником данных для различных приложений, в том числе информационного поиска, интеллектуального анализа текстов, конструирования и наполнения баз знаний. Для того чтобы неразмеченные PDF-документы стали доступны для машинной обработки, требуемой подобными приложениями, их недостающая структура должна быть восстановлена.

Анализ компоновки документов состоит в распознавании элементов структуры документа. Базовой частью этого процесса является сегментация страниц на блоки, которые затем можно классифицировать как заголовки, абзацы, ячейки таблиц, пункты списков и пр. Проблематика автоматического анализа компоновки документов активно развивается три последних десятилетия [2]. Тем не менее вопрос сегментации страниц неразмеченных PDF-документов остается открытым.

Известные алгоритмы сегментации страниц [3] предназначены в основном для работы с растровыми изображениями документов и печатно-ориентированным ASCII-текстом. По сравнению с этими форматами данных PDF предоставляет дополнительную информацию (порядок рендеринга, шрифтовые метрики, линейки и пр.), которая может улучшить качество анализа компоновки документов. В работе излагается опыт адаптации известных алгоритмов сегментации текста страниц изображений документов и ASCII-текста — T-Recs [4, 5], для того чтобы сделать их применимыми напрямую к формату PDF — неразмеченным случаям.

Наибольший интерес для применения данных алгоритмов представляет задача распознавания таблиц в неразмеченных PDF-документах [6, 7], в особенности в тех случаях, когда содержание ячеек — многострочные блоки текста. В работах [8, 9] показано, что эти алгоритмы могут эффективно применяться при оценке размещения текстовых блоков внутри кандидатных таблиц. В результате их применения возможно построить графовое представление кандидатной таблицы: вершины графа соответствуют текстовым блокам (т. е. абзацам текста вероятных ячеек таблицы), а ребра выражают расположение текстовых блоков относительно друг друга (т. е. размещение внутри вероятных строк и столбцов таблицы). На основе алгоритма “Случайный лес” была обучена модель бинарной классификации кандидатных таблиц на “истинные” и “ложные” по их графовому представлению [8]. Эта модель позволила отфильтровать ложноположительные случаи, появляющиеся при использовании некоторых искусственных нейросетевых моделей обнаружения таблиц в PDF-документах.

В статье рассматриваются основные подходы к сегментации страниц PDF-документов (разд. 1); вводятся предварительные определения (разд. 2); предлагается адаптация метода T-Recs к задаче сегментации страниц неразмеченных PDF-документов (разд. 3), включая модифицированный алгоритм начального объединения текстовых блоков и коррекции ложных случаев; в заключении приводятся выводы, кратко обсуждаются перспективы дальнейшего исследования.

1. Основные подходы

Выделяют три подхода к извлечению блоков текста из неразмеченных PDF-документов, используемых на практике:

- с конвертацией к редактируемому представлению;
- с конвертацией к растровым изображениям;
- без конвертации.

При реализации первого подхода исходные PDF-документы автоматически конвертируются в редактируемое представление, например ASCII-текст или HTML/XML-разметку. Текст считывается из подобного представления, при этом возможно выполнить объединение блоков текста, соответствующих отдельным словам, в блоки текста, соответствующие целым абзацам. Преимущество такого подхода связано с тем, что с помощью развитого программного обеспечения PDF-конвертации (например, pdf2text, `https://www.xpdfreader.com/pdftotext-man.html`, pdf2html, `http://pdftohtml.sourceforge.net`) удается перейти к человекочитаемому представлению, обеспечивающему более простое декодирование по сравнению с исходным PDF-форматом. Однако в общем случае такое конвертирование сопровождается ошибками извлечения "символов", "слов", "блоков текста", вносимыми применяемыми утилитами. В частности, это приводит к потере информации, которая могла бы использоваться при принятии решений в процессе анализа компоновки документов.

Второй подход предполагает предварительную растеризацию PDF-документов и дальнейшее извлечение таблиц уже из изображений документов. Следует отметить, что из трех перечисленных подходов этот является наиболее общим и делает возможным применение к задаче сегментации документов более общие техники компьютерного зрения. Известные методы, реализующие этот подход, традиционно базировались на правилах и машинном обучении. Основное направление их развития связано с применением глубокого обучения, в том числе моделей обнаружения объектов на изображениях, семантической сегментации изображений, а также генеративно-состязательных моделей.

При третьем подходе текст и графика извлекаются напрямую из исходных PDF-документов. Для того чтобы получить доступ к низкоуровневым инструкциям рендеринга текста и графики и собрать информацию, необходимую для сегментации страниц, используются различные инструментальные средства декодирования и воспроизведения PDF (например, PDFBox, `https://pdfbox.apache.org`, iText, `https://github.com/itext`). Преимуществом этого подхода является полнота исходной информации (текстовые позиции, шрифтовые метрики, порядок рендеринга, векторная графика и перемещение "пера" в процессе рендеринга), которую можно использовать для анализа компоновки страниц в целом и извлечения таблиц в частности. Их основной недостаток состоит в ограниченной применимости: их невозможно использовать в случае растрированных PDF-документов и некорректных кодировок встроенных глифов.

В настоящей работе предлагается остановиться на третьем подходе, для того чтобы улучшить качество сегментации страниц за счет вовлечения PDF-специфичной информации. Кроме того, предлагается адаптировать известный метод анализа компоновки изображений документов и печатно-ориентированного ASCII-текста — T-Recs [4, 5] к задаче сегментации страниц неразмеченных PDF-документов. Основной причиной выбора стало то, что метод T-Recs применим для случаев выравнивания текста по ширине. Именно такой тип выравнивания можно часто наблюдать в PDF-документах, в том числе в ячейках таблиц.

Следует отметить, что оригинальный метод T-Recs можно применить при реализации любого из трех подходов без изменений. Однако, в отличие от двух первых, третий подход позволяет расширить перечень фактов, на которые могут опираться адаптированные алгоритмы T-Recs. Именно это позволяет улучшить качество объединения текстовых блоков по сравнению с оригинальной версией алгоритмов T-Recs.

2. Предварительные определения

PDF-файл состоит из инструкций настройки графического контекста, рендеринга (т. е. вывода в графическом контексте) текста, векторной графики и растровых изображений. Один и тот же текст или графика могут быть представлены различными способами. Разные PDF-генераторы (виртуальные принтеры, конвертеры) для одного и того же источника (например, электронной таблицы) формируют разные на физическом уровне представления PDF-документы, не отличимые визуально, но имеющие отличные друг от друга наборы инструкций (рис. 1).

В процессе воспроизведения PDF-документа на графическом контексте можно извлечь исходную информацию, включая объекты трех видов:

- символьная позиция — символ, выводимый в некотором ограничивающем прямоугольнике с заданными шрифтовыми метриками;
- линейка разграфки — прямая линия, отрисованная между двумя точками ортогонально одной из осей координат (вертикально или горизонтально);
- след пера — прямая линия, соответствующая перемещению пера графического контекста из одной точки в другую без отрисовки линейки разграфки.

Известно, в каком порядке выводятся эти объекты на результирующее изображение документа. Это позволяет связать с каждой символьной позицией ее номер в общем порядке рендеринга.

Выходная информация данного этапа составлена из текстовых блоков, каждый из которых охватывает одну или несколько соседних символьных позиций. Примем следующую нотацию для представления позиций ограничивающего прямоугольника текста t как символьной позиции, так и текстового блока:

$$\operatorname{bbox} t = (x_l(t), y_t(t), x_r(t), y_b(t)) ,$$

где $x_l(t)$ — левая, $y_t(t)$ — верхняя, $x_r(t)$ — правая и $y_b(t)$ — нижняя границы соответственно. Используется нативная система координат PDF-документа, когда ось X возрастает слева направо, а ось Y — сверху вниз.

Определим текстовый блок b как $b = (C, \operatorname{bbox} b)$, где $C = \{c_1, \ldots, c_n\}$ — это набор соседних символьных позиций, а $\operatorname{bbox} b$ — это ограничивающий прямоугольник, охватывающий все символьные позиции из набора C так, что

$$x_l(b) = \min_{\forall c \in C} x_l(c), \quad y_t(b) = \min_{\forall c \in C} y_t(c), \quad x_r(b) = \max_{\forall c \in C} x_r(c), \quad y_b(b) = \max_{\forall c \in C} y_b(c).$$

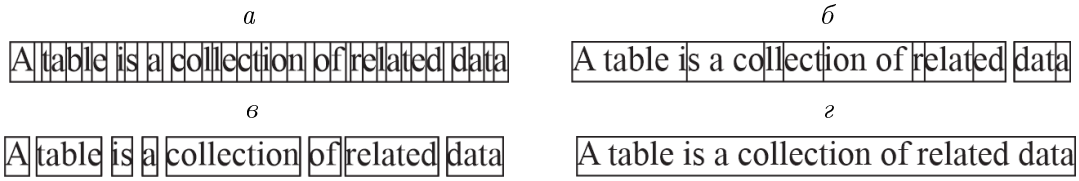

Рис. 1. Варианты генерации текста в PDF: текстовый блок — отдельный символ (*а*), несколько подряд идущих символов одной строки, любых (*б*) или только непробельных (*в*), целая строка (*г*)

Fig. 1. Options for text generation in PDF

3. Сегментация текста внутри страниц

Сегментация текста внутри страниц PDF-документов состоит в том, чтобы наиболее правдоподобно восстановить позиции абзацев.

Декодирование данных из PDF-документа требует его интерпретации, аналогичной его отображению на графическом устройстве. Интерпретация инструкций рендеринга текста и графики зависит от проигрываемого контекста устройства, определяющего стилевые характеристики (шрифт, цвета контура и заливки и пр.), и преобразования в системе координат. В качестве интерпретатора нами используется стековый процессор PDFBox. В результате считываются все символьные позиции, линейки и следы пера заданной страницы документа.

Прежде всего требуется объединить соседние непробельные символьные позиции в однокомпонентные текстовые блоки, соответствующие отдельным "словам". В силу своей тривиальности данный процесс не описывается в настоящей работе. В свою очередь соседние однокомпонентные текстовые блоки можно объединить в многокомпонентные, соответствующие уже целым "абзацам". Этот процесс реализован на основе восходящей сегментации страниц документов, выполняемой адаптированными алгоритмами T-Recs [4, 5]. Метод T-Recs предусматривает два этапа: начальное объединение блоков текста (от "слов" к "абзацам") и коррекцию ложных случаев. Нами был адаптирован алгоритм начального объединения блоков текста с учетом особенностей PDF-документов, а также предложено собственное решение коррекции ложных случаев.

3.1. Начальное объединение блоков текста

Оригинальная версия алгоритма начальной кластеризации "слов" T-Recs принимает в качестве входных данных "слова", для которых известны их символьные и строчные позиции в ASCII-тексте. Единственным условием объединения двух "слов" в один текстовый блок является то, что они должны быть размещены в соседних строках и пересекаться по символьным позициям. По сравнению с ASCII-текстом PDF-документы не имеют строчных и символьных позиций. Напротив, они предоставляют произвольное размещение на странице и форматирование текста. Поэтому данный алгоритм невозможно применить к PDF-документам напрямую, но возможно адаптировать.

Вход адаптированной версии алгоритма стоит из набора "слов" $W = \{w_1, \ldots, w_n\}$, для которых известны их ограничивающие прямоугольники на странице исходного документа, порядок рендеринга и шрифтовые метрики символьных позиций. Выходом алгоритма является набор текстовых блоков $B = \{b_1, \ldots, b_m\}$, $m \leq n$, образованных из "слов" набора W. Последовательность шагов рассматриваемого алгоритма представлена ниже.

Шаг 1. Выбрать случайное "слово" $w : w \in W$.
Шаг 2. Создать новый текстовый блок b в наборе B.
Шаг 3. Переместить "слово" w из набора W в текстовый блок b.
Шаг 4. Выбрать все "слова" $\tilde{W} \subset W$, являющиеся соседями "слова" w.
Шаг 5. Повторить рекурсивно шаги 3–5 для каждого "слова" $\tilde{w} : \tilde{w} \in \tilde{W}$.
Шаг 6. Перейти к шагу 1, пока набор W не пустой.

На шаге 4, для того чтобы найти соседние "слова" $\tilde{w}$, по ограничивающему прямоугольнику "слова" w определяется прямоугольная область поиска s следующим образом: $s(w) = (x_l(w), y_t(w) - h, x_r(w), y_b(w) + h)$, где h — это h-метрика, за

которую можно принять некоторую линейную функцию от межстрочного интервала. В нашем случае за h берется сумма среднего значения всех межстрочных интервалов на странице документа и высоты “слова” w. Поскольку межстрочный интервал не представлен в PDF-документе явным образом, оцениваем его значение приближенно по другим шрифтовым метрикам исходных символьных позиций. Следует отметить, что в силу ограничения на объем статьи не представляется возможным привести здесь алгоритмы оценки межстрочного интервала. Когда существует “слово” $\tilde{w}$, ограничивающий прямоугольник которого пересекается с областью поиска $s(w)$, то полагаем, что w и $\tilde{w}$ могут принадлежать одному текстовому блоку b.

Другим изменением, введенным нашей адаптацией, является проверка собираемых текстовых блоков. Предполагается, что каждый из них должен удовлетворять ряду дополнительных ограничений.

1. Порядок рендеринга его “слов” не должен разрываться.
2. Внутри его ограничивающего прямоугольника не должно быть линеек.
3. Между его “словами” не должно быть вертикальных следов пера.
4. Его “слова” должны иметь общие шрифты.

Первое ограничение базируется на предположении, что порядок чтения текста часто совпадает с порядком его рендеринга в PDF-документах, как показано на рис. 2. Это означает, что внутри одного “абзаца” символьные позиции, как правило, выводятся последовательно слева направо и снизу вверх. Номера в общем порядке рендеринга для смежных “слов”, расположенных в одной строке слева направо, будут отличаться на единицу. В случае многострочного текстового блока номер крайнего правого “слова” в строке сверху будет также отличаться на единицу от номера крайнего левого “слова” в смежной строке снизу.

Второе ограничение устанавливает, что линейки разграфки не могут пересекать текст внутри “абзаца”. Третье ограничение определяет, что, по крайней мере, вертикальные следы перемещения пера при их наличии совпадают с границами “абзаца”, и поэтому так же, как и “видимые” линейки, не могут пересекать текстовый блок. Четвертое ограничение сопоставляет шрифты символьных позиций, размещенных внутри собираемого текстового блока. Предполагается, что они должны относиться к одному семейству шрифтов, а их размеры не могут превышать разницу оцениваемых приближенно размеров строчного и под (над-)-строчного вариантов текста. В случае, когда все перечисленные ограничения выполнены, принимается окончательное решение об объединении “слов” в один текстовый блок.

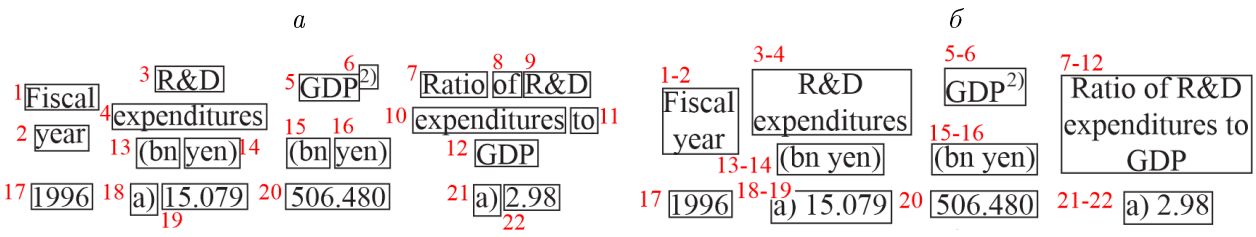

Рис. 2. Использование порядка рендеринга текста на странице PDF-документа (позиции вывода текстовых блоков представлены красным шрифтом): *а* — исходные “слова”; *б* — выходные “абзацы”

Fig. 2. Usage of the text rendering order on a PDF document page

3.2. Коррекция ложных случаев

Как в оригинальной, так и в адаптированной версии начальное объединение “слов” в блоки (см. подразд. 3.1) может приводить к ошибкам:

– ложноотрицательным, когда “слова” одной ячейки (абзаца) остаются необъединенными;
– ложноположительным, когда “слова” двух разных ячеек (абзацев) оказываются объединенными.

Метод T-Recs предлагает некоторые эвристики для коррекции ряда ложных случаев, неизбежно возникающих в результате применения T-Recs к ASCII-тексту и поэтому называемых “врожденными” ошибками [4, 5].

Следует отметить, что по сравнению с оригинальной версией метода T-Recs адаптированная версия, во-первых, использует более строгое условие объединения “слов”, а, во-вторых, рассчитана на другое представление входных данных (PDF вместо ASCII-текста). Поэтому она может приводить к другим ошибочным срабатываниям, не характерным для связки оригинального T-Recs- и ASCII-текстов. Нами предложено собственное решение коррекции ошибок, включающее три случая, рассматриваемых ниже.

3.2.1. Объединение изолированных текстовых блоков

Когда некоторое “слово” w совместно с другими “словами” в действительности составляет один “абзац”, но при этом не имеет соседей в своей области поиска $s(w)$, то в результате начального объединения оно окажется ложно изолированным в однокомпонентном текстовом блоке. Например, этот случай можно наблюдать при наличии однострочных абзацев, а также при выравнивании текста по краю или по центру (рис. 3).

С целью коррекции таких ошибок для каждого текстового блока b определяются две области поиска соседей следующим образом: $a_l(b) = (x_l(b) - s, y_t(b), x_r(b), y_b(b))$ и $a_r(b) = (x_l(b), y_t(b), x_r(b) + s, y_b(b))$, где s — это s-метрика, за которую можно принять некоторую линейную функцию от ширины пробела. В нашем случае s равна 3/2 от среднего значения пробельных ширин, считываемых из PDF-документа для всех символьных позиций внутри текстового блока b. Выбираются соседние текстовые блоки $\tilde{b}$, захватываемые любой из двух областей — $a_l(b)$ или $a_r(b)$. Если между парой b и $\tilde{b}$ нет линеек разграфки и вертикальных следов пера, то они объединяются в один блок. Приведенная процедура выполняется для всех таких пар соседних текстовых блоков.

3.2.2. Объединение пересекающихся текстовых блоков

Другой вид ошибок выражается в том, что ограничивающие прямоугольники текстовых блоков иногда пересекаются. Эта ситуация возникает, как правило, при использовании

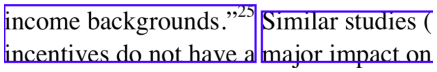

Рис. 3. Изолированные блоки текста
Fig. 3. Isolated blocks of the text

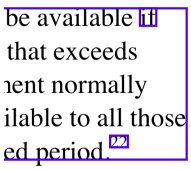

Рис. 4. Пересекающиеся блоки текста
Fig. 4. Intersecting blocks of the text

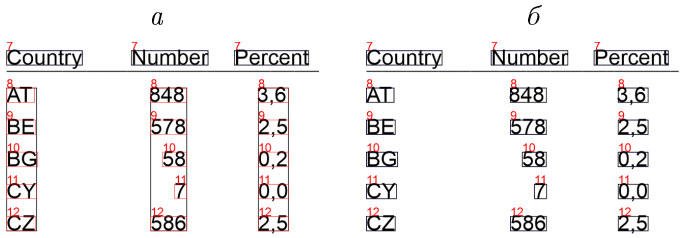

Рис. 5. Ложноположительные многострочные текстовые блоки (*а*) разделяются на однострочные (*б*) при обнаружении дупликации порядка рендеринга “слов”
Fig. 5 False positive multiline text blocks

различных шрифтов для форматирования разных частей текста одного “абзаца”. С другой стороны, такая ситуация не обязательно некорректна: часто она проявляется, когда размещение текста не соответствует так называемой манхэттенской компоновке, при которой все блоки текста обязательно вписаны в непересекающиеся прямоугольники. В данной работе принято решение пренебречь случаями неманхэттенской компоновки текста. В силу данного ограничения любые случаи пересечения ограничивающих прямоугольников текстовых блоков рассматриваются как ошибки (рис. 4). Эти случаи исправляются за счет объединения пересекающихся текстовых блоков.

3.2.3. Разделение текстовых блоков с дупликацией порядка рендеринга

Третий случай характерен для табличной компоновки текста — это “слова”, ложно объединенные в один текстовый блок, хотя в действительности они принадлежат разным строкам, но одному столбцу таблицы (рис. 5, *а*). Такие ошибки возникают при построчном выводе символьных позиций таблицы, который иногда можно наблюдать в некоторых PDF-документах. При этом “слова” соседних текстовых блоков при построчном сопоставлении будут иметь одинаковый порядок рендеринга. Данная эвристика служит для того, чтобы обнаружить такие ложноположительные многострочные текстовые блоки и разделить их на отдельные однострочные текстовые блоки (рис. 5, *б*).

Заключение

Исследования в области анализа компоновки документов имеют многолетнюю историю. За это время были предложены алгоритмы сегментации страниц (текста), представленных в виде растровых изображений документов и печатно-ориентированного ASCII-текста. Очевидно, что некоторые из этих алгоритмов могли бы быть адаптированы к задаче сегментации страниц неразмеченных PDF-документов без дополнительной конвертации исходных данных к изображениям или ASCII-тексту. Описанный в данной работе опыт показал, что использование PDF-специфичной информации (порядок рендеринга, шрифтовые метрики, линейки и пр.) позволяет улучшить качество сегментации текста адаптированными алгоритмами T-Recs. Преимущество данных алгоритмов заключается в том, что отсутствует необходимость в предварительной настройке параметров и обучении с учителем. В целом работа продемонстрировала возможность адаптации известного метода T-Recs к PDF-специфике.

Дальнейшее исследование может касаться вопросов применимости адаптированных алгоритмов к задачам распознавания не только таблиц, но и других текстовых элементов структуры документа: заголовков, абзацев, списков, надписей, колонтитулов и пр. Представленный опыт будет полезен специалистам, которые столкнулись с необходи-

мостью сегментации текста в неразмеченных PDF-документах. Возможно, некоторые из описанных находок найдут применение при адаптации к PDF-специфике также и других известных алгоритмов сегментации страниц документов, ориентированных изначально на растровые изображения или ASCII-текст.

Благодарности. Работа частично поддержана грантом № 075-15-2020-787 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект "Фундаментальные основы, методы и технологии цифрового мониторинга и прогнозирования экологической обстановки Байкальской природной территории").

Список литературы

[1] **Binmakhashen G., Mahmoud S.** Document layout analysis: a comprehensive survey. ACM Computing Surveys. 2020; 52(6):1–36. Article No.109. DOI:10.1145/3355610.

[2] **Kieninger T.** Table structure recognition based on robust block segmentation. Proceedings Volume 3305, Document Recognition V. 1998. DOI:10.1117/12.304642.

[3] **Kieninger T., Dengel A.** The T-Recs table recognition and analysis system. DAS'98: Selected Papers from the 3rd IAPR Workshop on Document Analysis Systems: Theory and Practice. 1998: 255–269.

[4] **Mikhailov A., Shigarov A., Rozhkov E., Cherepanov I.** On graph-based verification for PDF table detection. 2020 Ivannikov ISPRAS Open Conference. 2020: 91–95. DOI:10.1109/ISPRAS51486.2020.00020.

[5] **Mikhailov A., Shigarov A.** Page layout analysis for refining table extraction from PDF documents. 2021 Ivannikov ISPRAS Open Conference. 2021: 114–119. DOI:10.1109/ISPRAS53967.2021.00021.

[6] **Nganji J.T.** The Portable Document Format (PDF) accessibility practice of four journal publishers. Library & Information Science Research. 2015; 37(3):254–262. DOI:10.1016/j.lisr.2015.02.002.

[7] **Oliveira D.A.B., Viana M.P.** Fast CNN-based document layout analysis. Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops. 2017: 1173–1180.

[8] **Shafait F., Keysers D., Breuel T.M.** Performance comparison of six algorithms for page segmentation. LNCS 3872. Document Analysis Systems VII. 2006: 368–379. DOI:10.1007/11669487_33.

[9] **Shigarov A., Mikhailov A., Altaev A.** Configurable table structure recognition in untagged PDF documents. Proceedings of the 16th ACM Symposium on Document Engineering. 2016: 119–122.

Вычислительные технологии, 2022, том 27, № 5, с. 69–78. © ФИЦ ИВТ, 2022 ISSN 1560-7534
Computational Technologies, 2022, vol. 27, no. 5, pp. 69–78. © FRC ICT, 2022 eISSN 2313-691X

INFORMATION TECHNOLOGIES

DOI:10.25743/ICT.2022.27.5.007

Page text segmentation in untagged PDF documents

Shigarov Alexey O.*, Paramonov Viacheslav V.

Matrosov Institute for System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Russia
*Corresponding author: Shigarov Alexey O., e-mail: shigarov@icc.ru
Received June 08, 2022, accepted June 30, 2022.

Abstract

Currently, a large amount of non-editable documents are published and distributed in PDF (Portable Document Format). Often, they are "untagged", i. e. there are no annotation about their structure, including headings, paragraphs, tables, lists, figures, footers, etc. The document layout analysis consists in recognizing the listed elements of the structure. A basic part of this process is the segmentation of page text into blocks that can be classified as headings, paragraphs, table cells, etc. The well-known page segmentation algorithms are mainly designed to deal with either bitmap images of document pages or print-oriented ASCII text. Compared to these data formats, PDF provides additional information (rendering order, font metrics, ruling lines, etc.) that can improve document layout analysis. The paper describes our experience on the adaptation of some existing algorithms for segmenting page text in document images and ASCII text to make them applicable directly for PDF format — untagged cases.

Keywords: document layout analysis, document segmentation, document images, PDF accessibility, document information processing.

Citation: Shigarov A.O., Paramonov V.V. Page text segmentation in untagged PDF documents. Computational Technologies. 2022; 27(5):69–78. DOI:10.25743/ICT.2022.27.5.007. (In Russ.)

Acknowledgements. The work was partially supported by the Ministry of Science and Higher Education of the Russian Federation, the grant No. 075-15-2020-787 for implementation of Major scientific projects on priority areas of scientific and technological development (the project "Fundamentals, methods and technologies for digital monitoring and forecasting of the environmental situation on the Baikal natural territory").

References

1. **Binmakhashen G., Mahmoud S.** Document layout analysis: a comprehensive survey. ACM Computing Surveys. 2020; 52(6):1–36. Article No.109. DOI:10.1145/3355610.
2. **Kieninger T.** Table structure recognition based on robust block segmentation. Proceedings Volume 3305, Document Recognition V. 1998. DOI:10.1117/12.304642.
3. **Kieninger T., Dengel A.** The T-Recs table recognition and analysis system. DAS'98: Selected Papers from the 3rd IAPR Workshop on Document Analysis Systems: Theory and Practice. 1998: 255–269.
4. **Mikhailov A., Shigarov A., Rozhkov E., Cherepanov I.** On graph-based verification for PDF table detection. 2020 Ivannikov ISPRAS Open Conference. 2020: 91–95. DOI:10.1109/ISPRAS51486.2020.00020.
5. **Mikhailov A., Shigarov A.** Page layout analysis for refining table extraction from PDF documents. 2021 Ivannikov ISPRAS Open Conference. 2021: 114–119. DOI:10.1109/ISPRAS53967.2021.00021.
6. **Nganji J.T.** The Portable Document Format (PDF) accessibility practice of four journal publishers. Library & Information Science Research. 2015; 37(3):254–262. DOI:10.1016/j.lisr.2015.02.002.
7. **Oliveira D.A.B., Viana M.P.** Fast CNN-based document layout analysis. Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops. 2017: 1173–1180.
8. **Shafait F., Keysers D., Breuel T.M.** Performance comparison of six algorithms for page segmentation. LNCS 3872. Document Analysis Systems VII. 2006: 368–379. DOI:10.1007/11669487_33.
9. **Shigarov A., Mikhailov A., Altaev A.** Configurable table structure recognition in untagged PDF documents. Proceedings of the 16th ACM Symposium on Document Engineering. 2016: 119–122.