

РАСПОЗНАВАНИЕ СТРУКТУРИРОВАННЫХ ДОКУМЕНТОВ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ

С.В. Голубев,

аспирант кафедры «Распознавание изображений и обработки текста»
Московского физико-технического института.

Адрес: Московская область, г. Долгопрудный, Институтский пер., д. 9.

E-mail: sergey_g@abbyy.com.

В работе рассматривается задача распознавания структурированных документов на основе методов структурного распознавания образов. Предлагается система распознавания форм, включающая графовую модель документа для описания структуры печатных форм и метод построения обобщенной модели на основе обучающих примеров.

Ключевые слова: распознавание форм, машинное обучение, графовая модель документа.

1. Введение

С развитием информационных технологий и систем электронного документооборота важной задачей становится преобразование информации с бумажных носителей в электронную форму. Такое преобразование успешно осуществляется системами оптического распознавания символов. Однако часто посимвольное преобразование является недостаточным, и для перевода информации в электронное представление требуется распознавание логической структуры документа, что особенно важно при распознавании так называемых форм — анкет, бланков, различных финансовых документов.

В настоящее время для распознавания структурированных документов успешно применяются методы, основанные на структурном распознавании образов [1, 2, 3, 4, 9, 10]. Заметную трудность в применении методов структурного распознавания

образов представляет построение соответствующей модели документа. В некоторых случаях для решения этой проблемы применяются методы машинного обучения, позволяющие строить модель документа на основе некоторого количества изображений, размеченных вручную [11, 12, 13].

В данной работе рассматривается система распознавания форм ABBYY FlexiLayout [15], в которой используется специализированная модель документа (структурное описание). Модель описывает атрибуты структурных элементов, метрические и порядковые отношения между ними и с этой точки зрения сходна с графовыми моделями [2, 3]. Однако структурное описание FlexiLayout имеет ряд особенностей, в частности, для описания структуры используется специализированный язык, позволяющий вычислять параметры модели в процессе распознавания документа на основе текущих результатов. Это позволяет лучше адаптировать

модель к различным типам документов, в том числе со сложной структурой и наличием структурных вариаций (подробнее модель описана в разделе 3).

Существенным недостатком системы FlexiLayout является необходимость построения модели документа вручную, что весьма трудоемко и требует высокой квалификации пользователя. В данной работе предлагается метод построения структурных описаний для системы FlexiLayout путем обучения на примерах. Использование машинного обучения позволяет упростить процесс построения структурного описания документа, а использование в качестве модуля распознавания системы FlexiLayout дает возможность легко доработать результат обучения вручную для повышения точности распознавания документов со сложной структурой.

На этапе обучения используется промежуточная графовая модель документа, которая после обобщения преобразуется в структурное описание FlexiLayout. В статье рассматривается графовая модель и метод ее обобщения. Для обобщения графовой модели вводится специальная мера сходства графов. В разделе 2 рассмотрена задача распознавания структурированных документов, их основные разновидности и методы распознавания. В разделе 3 описывается модель документа FlexiLayout, используемая при распознавании, и графовая модель, используемая на этапе обучения. В разделе 4 рассматривается метод построения обобщенного (шаблонного) графа на основе обучающих примеров. В разделе 5 приводится экспериментальная оценка точности распознавания.

2. Распознавание структурированных документов

Выделяют четыре основных класса структурированных документов [16]:

- ◆ Жесткие формы
- ◆ Полужесткие формы
- ◆ Гибкие формы
- ◆ Документы произвольной структуры

Нужно отметить, что значительную часть структурированных документов составляют так называемые формы. Характерной особенностью форм является структура, состоящая из статической части, представленной заголовками, разделительными линиями и т.п., и внесенных в форму данных, которые будем называть полями. Примерами форм могут служить анкеты, финансовые документы типа счетов-фактур, платежных поручений и т.п. Задача

распознавания структурированных документов состоит в том, чтобы, имея изображение документа (как правило, полученное в результате сканирования), определить локализацию требуемых данных (в случае форм это поля).

Жесткие формы имеют фиксированное расположение полей, поэтому их распознавание не требует серьезного анализа структуры и текста. Достаточно исправить искажения, полученные при сканировании и использовать шаблон, в котором заданы координаты полей.

Полужесткие формы являются аналогом жестких форм, в которых сняты некоторые ограничения, налагаемые на разметку. Примером таких документов являются формы, заполняемые в электронном виде с последующей распечаткой, которые позволяют сдвигать последующие поля, если требуется дополнительное место для данных в текущем поле.

В гибких формах имеются значительные вариации в расположении полей и статических элементов. Однако, как правило, для каждого поля можно выделить связанный с ним локальный контекст, который может содержать заголовок поля, несколько разделительных линий и т.п. При этом расположение поля относительно контекста меняется мало.

Наконец, самой общей категорией являются документы произвольной структуры. Сложность их распознавания может достаточно сильно отличаться, в сложных случаях может требоваться анализ естественного языка.

Для распознавания форм успешно применяются методы на основе структурного распознавания образов. Широкое распространение получила графовая модель документа, в которой текстовые блоки и разделительные линии образуют узлы графа, а ребра графа соответствуют отношениям между ними [1, 2, 3, 4]. Распознавание в данном случае состоит в сопоставлении модельного графа и графа, полученного по изображению документа. Простые варианты графовых моделей — графы отношений соседства или порядка достаточно просты, чтобы их можно было построить автоматически. Модели с метрическими отношениями задают взаимные ограничения на координаты структурных элементов, что приводит к более предсказуемым результатам распознавания. Для придания большей гибкости по нескольким экземплярам определяются средние расстояния, а возможные отклонения задаются со значительным запасом. Дальнейшим развитием графовых моделей с метрическими отношениями являются модели на

```

KeywordElement DateHeader
  SearchText: "Invoicedate: | Invoicedate";
  MaxGapInLine: 50 * dot;
EndOfElement

DateElement Date
  DateFormat: DayMonthYear;
  Language: "English";
  MinDate: "14/09/1752";
  MaxDate: "31/12/9999";

  let Header = InvoiceGroup.DateHeader;
  if not Header.IsNull then {
    let rect1 = Rect (Header.Rect.Right, Header.Rect.Top -20dt,
      PageRect.Right, Header.Rect.Bottom+20dt);
    let rect2 = Rect (Header.Rect.Left - 200dt, Header.Rect.Bottom,
      Header.Rect.Right + 150dt, Header.Rect.Bottom+200dt);

    RectArray ar;
    ar = RectArray( rect1 );
    ar.Add( rect2 );
    RestrictSearchArea( ar );
  } else {
    Above: PageRect.Top + PageRect.Height/2;
  }
EndOfElement

KeywordElement InvoiceHeader
  SearchText: "Invoice";
  MaxGapInLine: 50 * dot;
  NearestY: PageRect.Top;

  Above: SearchElements.InvoiceGroup.DateHeader, 0 * dot;
  Exclude: SearchElements.InvoiceGroup.DateHeader;
EndOfElement

LetterChainElement InvoiceNumber
  MaxGapInLine: 30 * dot;
  Alphabet: "0123456789", 1, true;
  MaxErrorsCount: 0.2;
  TotalChainLength: -7, 3, 5, 15;
  Nearest: Header;

  let Header = InvoiceGroup.InvoiceHeader;
  if not Header.IsNull then {
    let rect = Rect (Header.Rect.Right, Header.Rect.Top -20dt,
      PageRect.Right, Header.Rect.Bottom+20dt);

    RestrictSearchArea( rect );
  } else {
    Above: PageRect.Top + PageRect.Height/2;
  }
EndOfElement

```

INVOICE 10560

Client 3420150

Invoice date: 10 - 09 - 2004

статические элементы



поля

Рис. 1. Фрагмент платежного документа.

основе случайных [5] и нечетких графов [6]. В первом случае в модели задаются распределения вероятностей для атрибутов вершин и ребер и при распознавании вычисляется вероятностная оценка графа, полученного по изображению, которая определяет степень его соответствия модели. В случае нечетких графов вместо диапазонов значений атрибутов на ребрах и вершинах используются нечеткие множества значений. Использование нечетких множеств позволяет лучше описывать возможные вариации атрибутов структурных элементов и их взаимного расположения.

Следует отметить методы для распознавания табличных документов. Для их идентификации применяются различные способы сегментации таблицы на ячейки, по которым затем определяется логическая структура [7, 8]. Для описания структуры таблиц так же могут применяться графовые модели [9, 10].

Помимо графовых моделей, для описания структуры документов применяются системы логических правил [11, 13]. В этом случае для обучения системы могут использоваться существующие системы логического вывода, например система INTHELEX [11].

Кроме методов на основе структурного распознавания образов следует отметить также системы распознавания, основанные на статистическом подходе [12, 13, 14].

3. Модель документа

В качестве модуля распознавания применяется система ABBYY FlexiLayout. Данная система основана на методах структурного распознавания образов. Однако, в отличие от систем на основе графовых моделей, в этой системе используется специализированная модель документа (структурное описание), позволяющая достичь большей гибкости по сравнению с простыми графовыми моделями (подробнее о структурных описаниях можно прочесть в работе [15]). Модель FlexiLayout описывает множество структурных элементов, соответствующих реквизитам документа. Для каждого элемента задается множество допустимых значений его атрибутов, позволяющее локализовать структурный элемент на изображении. При этом в отличие от других систем отсутствует априорное выделение структурных элементов для построения графа документа. Вместо этого структурные элементы выделяются непосредственно в процессе распознавания на основе структурного описания. Также в модели задается взаимное расположение эле-

ментов в виде порядковых и метрических отношений, отношений соседства и т.д., при этом могут быть заданы как обычные, так и нечеткие отношения. В случае использования обычных отношений модель схожа с графом метрических отношений. Для описания отношений и характеристик структурных элементов используется специализированный язык, который предоставляет достаточно широкие возможности в задании свойств элементов и их взаимного расположения. Для каждого элемента описание содержит интерпретируемый код инициализации параметров и отношений, который выполняется во время распознавания. За счет этого достигается возможность вычисления характеристик структурных элементов и отношений между ними «на лету» в процессе сопоставления модели документа и изображения. Например, допустимое расстояние между элементами может быть вычислено по размерам одного из них, и, следовательно, будет различным для разных экземпляров документа (например, если используется шрифт разного размера).

В качестве примера приведем структурное описание для фрагмента формы, изображенной на *рис. 1*:

Однако на этапе обучения работать непосредственно с моделью FlexiLayout не очень удобно, в частности из-за его текстового вида, поэтому желательно использовать более простую модель документа. В качестве промежуточного представления структуры документа используется граф метрических отношений. Это позволяет опираться на уже известные методы работы с данными, представленными графами [17, 18, 19]. Множество вершин модельного графа соответствуют графическим объектам в случае описания изображения или элементам логической структуры в случае обобщенного описания.

Вершины графа имеют метки вида $I_V = (I, D)$, где:

I — идентификатор объекта или структурного элемента, определяющий его логическую роль в документе, который может принимать значения из множества $\{R, S, F_1, F_2, \dots, F_N\}$, где:

F_i — идентификаторы полей. Поскольку при составлении обучающей выборки множество полей известно (определяется типом документа) и расположение полей указывается в обучающих примерах, то для графических объектов, представляющих поля, известны уникальные идентификаторы.

R — метка текстового объекта.

S — метка разделительной линии.

D — описание множества графических объектов, которыми элемент логической структуры представ-

ляется на изображении. Существует несколько способов описания элемента логической структуры:

♦ Множество, состоящее из одного графического объекта.

♦ Множество фраз из одного или нескольких слов. Описание имеет вид: $\{p_1, \dots, p_n\}$, где p_i — символные строки.

♦ Множество строк, состоящих из символов заданного алфавита или нескольких алфавитов $\{a_1, \dots, a_n\}$, где $a_i = (\{s_1, \dots, s_m\}, P)$, s_i — символы, P — максимальная доля символов алфавита в строке.

♦ Множество фрагментов текста, имеющих определенные характеристики: число строк, ширину и высоту фрагмента, наибольший пробел между словами в строке, допустимое расстояние между строками, выравнивание.

♦ Множество разделительных линий с определенными размерами.

Ребра графа соответствуют метрическим отношениям между объектами. Метки ребер имеют вид: $l_E = (d_x, d_y)$, где d_x, d_y — расстояние между прямоугольниками по горизонтали и вертикали в виде интервалов $(d_{\min}, d_{\max})$. Расстояния измеряются по ближайшим точкам, при этом если прямоугольники пересекаются, то соответствующее расстояние равно нулю. Расстояние может быть как положительным, так и отрицательным, и зависит от порядка вершин, т.е. ребра являются направленными.

Рассмотренная модель позволяет описывать как конкретное изображение (экземпляр документа) так и его обобщенную структуру. В первом случае будем говорить о *графе документа*, а во втором — о *шаблонном графе* или *обобщенном графе*.

4. Обобщение модели документа

Введем следующее определение. Будем считать, что граф документа G_D описывается шаблонным графом G_T , если существует отображение $X: V(G_T) \rightarrow V'(G_D)$, где $V'(G_D) \subseteq V(G_D)$ такое, что для любых вершин v и u и их образов v' и u' выполняются условия:

1. $I_v = I_{v'}$, т.е. идентификаторы вершин совпадают.
2. Объект, заданный вершиной v' , входит в множество объектов, заданное описанием D метки вершины v .
3. Метка l ребра $e = (v, u)$ является обобщением метки l' ребра $e' = (v', u')$, в том смысле, что интервалы d_x и d_y метки l входят в интервалы d_x' и d_y' метки l' .

Построение шаблонного графа будем проводить, последовательно обобщая его на документы из обучающей выборки. Чтобы обобщить шаблонный граф G_T на граф документа G_D , необходимо построить отображение подмножества вершин шаблонного графа на подмножество вершин графа документа: $Y: V'(G_T) \rightarrow V'(G_D)$, где $V'(G_T) \subseteq V(G_T)$, $V'(G_D) \subseteq V(G_D)$, такое, что $I_v = I_{v'}$. После этого метки вершин из множества $V'(G_T)$ должны быть обобщены с метками вершин из множества $V'(G_D)$ так, чтобы заданные метками на вершинах множества графических объектов включали новые объекты, описанные метками на вершинах $V'(G_D)$. Аналогично, интервалы расстояний на ребрах графа G_T должны быть обобщены с интервалами на образах этих ребер из графа G_D , соответствующие интервалы при этом объединяются:

$$(d_{\min}^1, d_{\max}^1) \cup (d_{\min}^2, d_{\max}^2) = (\min(d_{\min}^1, d_{\min}^2), \max(d_{\max}^1, d_{\max}^2)).$$

Очевидно, что может быть построено большое число таких отображений и соответственно шаблонных графов. Необходимо ввести критерий, позволяющий выбирать наиболее подходящий граф. Следует отметить, что рассмотренное отображение вершин представляет собой путь редактирования (edit path) [17]. В общем случае для оценки пути редактирования может быть применена вероятностная оценка [20], однако в данном случае более предпочтительно использование специальной эмпирической оценки, соответствующей предметной области.

Введем оценку качества отображения вершин и соответствующего обновленного шаблонного графа G_T' :

$$Q = \left(\prod Q_V(v_i \rightarrow u_j) \right) \cdot Q_U(V(G_T) \setminus V'(G_T)) \cdot Q_U(V(G_D) \setminus V'(G_D)),$$

где Q_V — оценка пары вершин из отображения Y , Q_U — оценка вершин из множеств $V(G_T) \setminus V'(G_T)$ и $V(G_D) \setminus V'(G_D)$, то есть вершин, оставшихся без образа или прообраза. Поскольку при отображении вершин идентификаторы вершин сохраняются, то соответствие между полями документа задано однозначно, поэтому в оценке шаблонного графа следует учитывать только вершины с метками текстовых объектов и разделительных линий.

$$Q_V(v \rightarrow u) = Q_V(v') = (1 - \alpha) \cdot \max_i Q_E(e_i) + \alpha \cdot \sum_i Q_E(e_i) / N_F$$

где $Q_E(e_i)$ — оценка i -го ребра, выходящего из вершины v' графа G_T' , полученной в результате отображения вершины v графа G_T на вершину u графа G_D ; $i \in [1, N_F]$; N_F — число полей в данном типе документа; α — эмпирический коэффициент, вводящий зависимость оценки не только от оценки наилучшего ребра, но и от средней оценки.

Оценка Q_U зависит от числа вершин в соответствующих множествах и имеет вид: $Q_U(V) = U^{|V|}$, т.е. за каждую не отображенную вершину назначается фиксированный штраф U .

Оценка ребра имеет вид:

$$Q_E(e) = \max(0, 1 - \left(\frac{\Delta_x + \Delta_y + 4 - \sqrt{(\Delta_x - \Delta_y)^2 + 16}}{2} \right)^2),$$

где $\Delta = (d_{max} - d_{min}) / W$, (d_{min}, d_{max}) — интервал d_x или d_y метки ребра e_i , W — характерная ширина интервала порядка размера страницы.

Данная эмпирическая оценка выбрана из следующих соображений. Во-первых, изолинии функции должны быть вогнутыми, т.е. оценка ребра с интервалами шириной (для примера) $\Delta_x = 1$, $\Delta_y = 0$ должна быть выше, чем для ребра с интервалами шириной $\Delta_x = \frac{1}{2}$, $\Delta_y = \frac{1}{2}$. Во-вторых, оценка должна быть близка к 1 для ребра, полученного в результате правильного сопоставления объектов (Δ_x и Δ_y при этом существенно меньше 1), и заметно снижаться при приближении значений Δ_x и Δ_y к 1. В данном случае изолиниями являются гиперболы вида $y = \frac{4}{x+a} - a$, $a \in [1, 2]$.

График функции оценки ребра в области $\Delta_x \in [0, 1]$, $\Delta_y \in [0, 1]$ представляет собой поверхность, полученную «скольжением» гиперболы $y = \frac{4}{x+1} - 1$, $z = 0$ по параболу $z = 1 - x^2$, $x = y$.

Для поиска отображения вершин с наилучшей оценкой применяется следующий метод:

♦ Будем последовательно выбирать пары для вершин первого графа, используя дерево перебора. Узел дерева соответствует решению отобразить

вершину v_i на вершину u_j , либо оставить вершину v_i без образа.

♦ Для каждого нетерминального узла и соответственно пути в дереве введем частичную оценку Q_p , которая, в отличие от Q , не учитывает нерассмотренные вершины первого графа и не имеющие прообраза вершины второго. Т.е. будем учитывать в $Q_U(V(G_T) \setminus V'(G_T))$ только рассмотренные вершины шаблонного графа, а $Q_U(V(G_D) \setminus V'(G_D))$ опустим. Не сопоставленные вершины будем учитывать только для конечных узлов дерева перебора, соответствующих построенному отображению Y .

♦ Перебор дерева производится в порядке уменьшения качества вершин. Поскольку при спуске по дереву оценка не может увеличиваться, то оценка текущей вершины одновременно является оценкой сверху всех продолжений данного пути, таким образом, порядок перебора соответствует алгоритму Дейкстры для дерева.

Рассмотренным образом шаблонный граф обобщается последовательно на экземпляры документа из обучающей выборки. Затем графовая модель преобразуется в структурное описание FlexiLayout. По вершинам графа строятся описания структурных элементов. При этом, поскольку метки на вершинах были специальным образом подобраны, описания структурных элементов получаются напрямую из меток на вершинах.

Ребра, описывающие отношения, проходят несколько фильтров перед преобразованием в отношения структурного описания. Дело в том, что модельный граф является полным, и при переносе всех заданных в нем метрических отношений в структурное описание, последнее окажется перегруженным, а кроме того не будет обладать достаточной гибкостью (аналог переобучения). Чтобы этого не произошло, отбираются только наиболее важные отношения. Для каждого поля выбираются несколько статических элементов, относительно которых поле локализуется наиболее надежно. С выбранными элементами строятся метрические отношения, ограничивающие положение поля. Ребра, связывающие статические элементы друг с другом, преобразуются в отношения порядка (поскольку метки на ребрах задают диапазоны расстояний с соответствующим знаком, то по такой метке можно построить как метрическое отношение, так и отношения порядка). Задание таких отношений позволяет повысить точность распознавания по сравнению с поиском только на основе локального контекста.

После того как структурное описание построено оно тестируется на обучающей выборке. Проверка необходима, поскольку некоторые структурные элементы могут распознаваться неоднозначно. Например, одно и то же ключевое слово может встречаться в нескольких местах страницы. На основе анализа таких ложных объектов в структурное описание вносятся необходимые изменения. Во-первых, может быть изменен порядок поиска структурных элементов с тем, чтобы в первую очередь были распознаны наиболее надежные статические элементы. Во-вторых, в описание могут быть добавлены дополнительные метрические отношения между статическими элементами, а так же некоторые дополнительные ограничения.

5. Экспериментальная оценка и выводы

Оценка обобщающей способности предложенного метода распознавания документов проводилась методом скользящего контроля при нарастающей длине обучающей выборки. Множество изображений документов сортировалось в случайном порядке. На каждой итерации в качестве обучающей выборки использовались первые n изображений, а в качестве контрольной — $(n+1)$ - е изображение. При этом определялось число ошибок, т.е. число полей, локализация которых была определена неверно. Проверка проводилась на нескольких различных сортировках исходного множества, после чего число ошибок для

обучающей выборки длины n усреднялось. На рис. 2 показана зависимость среднего процента ошибок от числа документов, на которых проводилось обучение.

Рассмотренная модель документа и основанная на ней система распознавания позволяют с достаточной точностью извлекать информацию из жестких и полужестких форм. Также система может использоваться для распознавания форм с гибкой структурой, однако в этом случае требуется больший размер обучающей выборки, а процент ошибок достаточно высок. Однако в этом случае возможна доработка полученного автоматически структурного описания вручную оператором. Такая доработка возможна благодаря тому, что модель FlexiLayout изначально предназначалась для построения вручную. Существует специализированная программа для ее редактирования, включающая набор необходимых для этого средств и инструментов. Даже с условием ручной доработки использование системы машинного обучения позволяет существенно сократить время и трудоемкость создания структурного описания.

Для документов произвольной структуры, в которых нельзя выделить статическую часть, например писем, данная система имеет ограниченное применение. Однако она может применяться для некоторых документов, не относящихся к формам, например контрактов, где некоторые ключевые слова (заказчик, адрес, телефон, поставщик и т.п.) могут быть использованы в качестве опорных элементов для локализации необходимых полей. ■

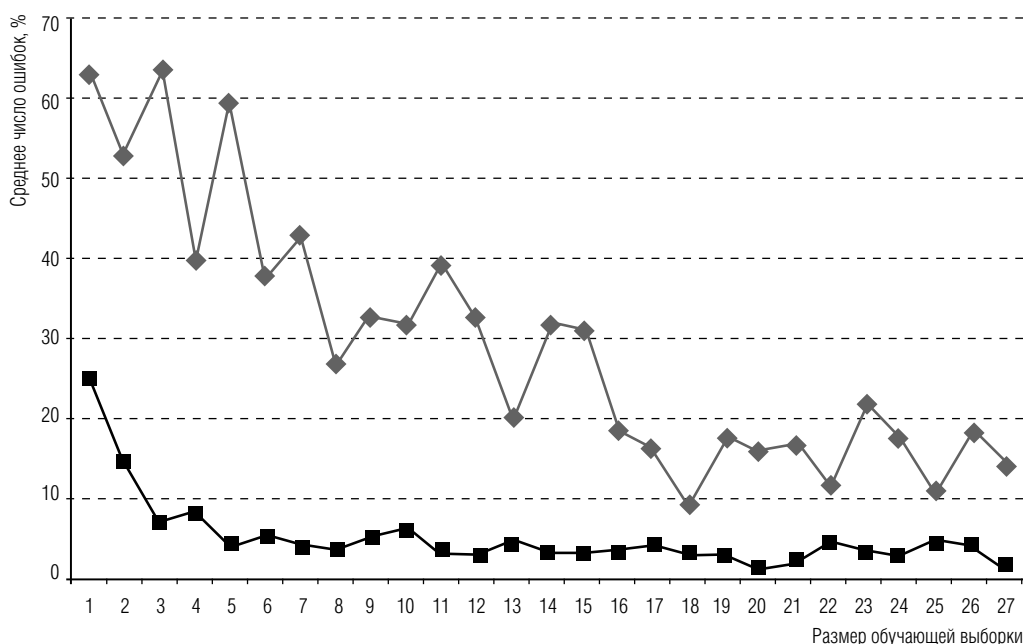


Рис. 2. Черный график — полужесткая форма, серый график — гибкая форма.

Литература

1. Farrow G.S.D. et al, Model Matching in Intelligent Document Understanding, Proc. of ICDAR'95, c. 293-296, 1995.
2. Ishitani Y., Flexible and robust model matching based on association graph for form image understanding, Pattern Analysis and Application, т. 3, с. 104–119, 2000.
3. Yuan J., Tang Y.Y., Suen C.Y., Four Directional Adjacency Graphs (FDAG) and Their Application in Locating Fields in Forms, Proc. of ICDAR'95, 1995.
4. Bayer, Th. A. and H. U. Mogg-Schneider, A Generic System for Processing Invoices, Proc. of ICDAR'97, 1997.
5. Bagdanov A. and Worring M., Fine-Grained Document Genre Classification Using First Order Random Graphs, Proc. of ICDAR'2001, c. 79-85, 2001.
6. Chan K. P., Learning Templates from Fuzzy Examples in Structural Pattern Recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26(1), c. 118-123, 1996.
7. Hirayama Y., Analyzing Form Images by Using Line-Shared-Adjacent Cell Relations, Proc. of IAPR'96, 1996.
8. Ramel J., Crucianu M., Vincent N. and Faure C., Detection, Extraction and Representation of Tables, Proc. of ICDAR'03, c. 374-379, 2003.
9. Hurst M., A Constraint-Based Approach to Table Structure Derivation, Proc. of ICDAR'03, c. 911-916, 2003.
10. Amano A. and Asada N., Graph Grammar Based Analysis System of Complex Table Form Document, Proc. of ICDAR'03, c. 916-921, 2003.
11. Esposito F., Ferilli S., Basile T. and Di Mauro N., Machine Learning for Digital Document Processing: from Layout Analysis to Metadata Extraction, Machine Learning in Document Analysis and Recognition, c. 105-108, Springer-Verlag, 2008.
12. Laven K., Roweis S. and Leishman S., A Statistical Learning Approach to Document Image Analysis, Proc. of ICDAR'05, c. 357-362, 2005.
13. Ceci M., Berardi M., and Malerba M., Relational Learning Techniques for Document Image Understanding: Comparing Statistical and Logical Approaches, Proc. of ICDAR'05, c. 473-478, 2005.
14. Minagawa A., Fujii Y., Takebe H., and Fujimoto K., Logical Structure Analysis for Form Images with Arbitrary Layout by Belief Propagation, Proc. of ICDAR'07, c. 714-719, 2007.
15. Tuganbaev D., Pakhchanian A., and Deryagin D. Universal Data Capture Technology from Semi-structured Forms, Proc. of ICDAR'05, c. 458-463, 2005.
16. Wnek J., Machine Learning of Generalized Document Templates for Data Extraction, Lecture Notes in Computer Science, Vol. 2423/2002, c. 627-635, 2002.
17. Cook D., Holder L., Mining Graph Data, Wiley-Interscience, 2006.
18. Kuramochi M., Karypis G., An efficient algorithm for discovering frequent subgraphs, Tech. Rep. 02-026 Minneapolis, University of Minnesota, 2002.
19. Yan X., Han J., gSpan: Graph-Based Substructure Pattern Mining, Proc. of IEEE International Conference on Data Mining (ICDM'02), Los Alamitos, 2002.
20. Neuhaus M. and Bunke H., A probabilistic approach to learning costs for graph edit distance. Proceedings of 17th International Conference on Pattern Recognition, Vol. 3, 2004.