УДК 004.89, 004.912 Дата подачи статьи: 10.06.16
DOI: 10.15827/0236-235X.116.107-112 2016. Т. 29. № 4. С. 107–112

МЕТОД РАСПОЗНАВАНИЯ СТРУКТУРЫ ТАБЛИЦЫ В ЭЛЕКТРОННЫХ ТАБЛИЧНЫХ ДОКУМЕНТАХ

Н.М. Ткешелашвили, аспирант, nmtkeshelashvili@corp.ifmo.ru;
С.В. Клименков, ассистент;
А.М. Дергачев, к.т.н., доцент
(Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО), Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)

Одним из популярных средств хранения деловой информации являются электронные таблицы. К сожалению, информация в них плохо структурирована. Определение структуры таблицы необходимо для корректного извлечения из нее данных в процессе автоматической обработки. В работе предложен метод распознавания структуры таблицы, основанный на визуальном подходе. Он опирается на тот факт, что в момент создания электронной таблицы заголовки, данные и агрегатные ячейки представляются таким образом, чтобы человек мог без проблем отличить их друг от друга. В разработанном методе типы данных и свойства форматирования ячеек представляются в виде набора битовых карт, рассматриваемых как графическое представление таблицы. Полагаясь на визуальные различия, позволяющие человеку отличать одни структурные элементы таблицы от других, а также на статистические зависимости внутри битовых карт, метод определяет ориентацию таблицы, расположение заголовков и данных и формирует структуру в виде набора объектов. Для определения направления таблицы используется метод Хафа. Для проверки корректности распознавания был выбран набор тестовых электронных таблиц, содержащих деловую информацию. Метод показал высокую точность: из более чем 100 тестовых файлов, содержащих более 20 000 строк, корректность распознавания составила 92 %.

Ключевые слова: *автоматическая обработка документов, распознавание структуры, электронные таблицы.*

Электронный документооборот применяется практически повсеместно, и одним из распространенных форматов документов являются электронные таблицы. Автоматическая обработка такого типа документов помогла бы решить многие трудоемкие задачи, связанные с извлечением информации, сопоставлением и сравнением данных из нескольких документов, построением и заполнением БД на основе электронных таблиц и так далее. Ключевой проблемой в данном случае является распознавание структуры таблицы, то есть определение таких ее частей, как заголовки, подзаголовки и непосредственно сами данные. Решению описанной проблемы посвящен ряд работ.

Решение, предложенное в [1], опирается на анализ содержащихся в таблице формул. Авторами вводится язык описания электронных таблиц и решается задача обнаружения структуры при помощи логического программирования на языке Prolog. В работе [2] рассматриваются поиск скрытых структурных свойств электронных таблиц и то, как эти свойства можно использовать для автоматических подсказок пользователю при редактировании электронных таблиц. Авторы ищут функциональные зависимости между наборами данных, по которым строится реляционная схема документа. В статье [3] описан метод плавного перехода между электронными таблицами и реляционной БД. Основой данного метода является гипотеза, что электронные таблицы и БД связаны отношениями уточнения. Для поиска таких отношений авторы ищут функциональные зависимости между данными. Использование вычислительных формул из функциональных зависимостей позволяет сформулировать уточняющие шаги. Подход авторов использует теорию уточнения данных [4].

Описанные методы с разных сторон подходят к задаче распознавания структуры таблицы. Некоторые из них требуют дополнительных данных о структуре, составленных вручную, другие опираются на связь данных через формулы. Предлагаемый в настоящей работе метод основан на визуальном подходе к анализу структуры таблицы. Он опирается на графический образ таблицы, полученный из исходного файла электронных таблиц, а не на ее строгий макет. Исходными данными являются табличные документы форматов XLS и XLSX. Метод не предназначен для распознавания таблиц из сканированных или иных графических изображений.

Понятие структуры таблицы

Рассмотрим структуру таблицы с точки зрения нормативных источников (ГОСТ 2.105-95). Согласно государственному стандарту, в таблице выделяют головку, боковик и матричное ядро. Головка таблицы содержит заголовки столбцов, которые могут быть многоуровневыми, но должны составлять иерархию. Боковик содержит заголовки строк. Возможно использование врезанных заголовков. Над таблицей располагаются ее название и номер, ниже может размещаться иная дополнительная информация, например, дата создания, имя автора или единая единица измерения величин. Государственный стандарт регламентирует пра-

вила оформления таблиц, но никак не затрагивает данные, хранящиеся в таблице.

Другой подход заключается в формальном описании таблиц с позиции информационных объектов, представляющих собой совокупность аргументов и зависимых переменных [5]. Этот подход близок к реляционной модели, согласно которой строка таблицы представляется набором смысловых отношений между ячейками [6]. В такой модели таблица хранит информацию о множестве объектов с определенными именованными свойствами (атрибутами). Каждая запись, таким образом, может быть преобразована в пару «атрибут–значение». Названия свойств указываются в заголовках столбцов, а значения в ячейках. Значения свойств имеют одинаковый тип. Объект занимает одну строку таблицы. В документах свойства могут объединяться в иерархии, формируя различные варианты визуального представления одного и того же отношения.

Метод извлечения структуры таблицы

Таблицы, как правило, содержат объекты одного типа, расположенные по строкам или по столбцам в зависимости от ориентации таблицы. Построчное представление более распространено, постолбцовое типично для акцентирования внимания на сравнении свойств объектов при их небольшом количестве. В случае ориентации по строкам одна строка описывает один объект, а заголовки содержат названия свойств, общих для всех объектов. Таким образом, корректно определив местоположение заголовков и данных, то есть решив задачу обнаружения структуры таблицы, можно извлечь объекты в виде набора свойств и их значений. Распознавание структуры таблицы состоит из следующих шагов:

шаг 1 – определение границ рабочей области таблицы;

шаг 2 – определение направления данных;

шаг 3 – поиск шаблона данных и заголовков;

шаг 4 – обработка заголовков и врезанных заголовков.

Шаг 1. Определение границ рабочей области таблицы. В документах электронных таблиц для обозначения одной страницы документа применяется понятие «лист» (sheet). В общем случае на одном листе могут помещаться несколько таблиц, но рассматриваемый метод предполагает наличие на каждом листе только одной таблицы.

При создании нового документа электронных таблиц редактор отображает пустой лист определенного по умолчанию размера, полностью заполненный ячейками. Однако фактически ячейки создаются в момент их редактирования пользователем. Таким образом, документ не содержит ячейки, которые не были модифицированы, сокращая этим размер хранимых данных. Тем не менее в некоторых случаях создаются ячейки, не несущие полезной информации. Например, при редактировании ячейки, находящейся не в начале строки, будут созданы все ячейки от начала строки до этой ячейки. То есть, если таблица размещена на листе с некоторым смещением от его краев, это место будет заполнено пустыми ячейками. Также при выделении целиком столбца и назначении ему форматирования создаются все ячейки столбца в соответствии с высотой листа, принятой по умолчанию.

Под рабочей областью таблицы понимается диапазон ячеек, содержащих полезную информацию и подлежащих обработке. В связи с особенностями, указанными выше, на первом шаге сокращается рассматриваемое множество ячеек за счет отбрасывания пустых. Рабочая область определяется как минимальный прямоугольный диапазон размером R_w на R_h, включающий все непустые ячейки листа.

Шаг 2. Определение направления данных. Таблицу, заголовки которой расположены над столбцами, называют столбцовой; если заголовки находятся в строках, таблица называется строчной. Под направлением данных понимается расположение объектов в таблице: по строкам или по столбцам.

Метод использует визуальные особенности структуры таблицы, такие как взаимное расположение элементов структуры, форматирование и типы данных ячеек. Каждая ячейка обладает набором атрибутов, относящихся как к самой ячейке, так и к данным внутри нее. Данные имеют формат записи и тип, к примеру, тип данных «числовой» и формат записи «проценты». Ячейка и данные могут иметь форматирование, к которому относятся цвет текста и фона, границы и отступы, выравнивание и пр. Каждому атрибуту соответствует набор возможных значений, например, атрибут «шрифт» может быть Arial, Verdana, Georgia и т.д.

На данном шаге для каждого значения каждого атрибута строится битовая карта. Размеры битовой карты совпадают с размерами рабочей области, единица означает наличие у данной ячейки атрибута с данным значением. На рисунке 1 приведен пример таблицы с информацией о книгах, имеющихся в книжном магазине.

Справа от таблицы приведен пример битовых карт для атрибута «тип данных». На первой карте он принимает значение «текст», на второй – «числовой», на третьей – «дата». Множество битовых карт, сформированных для всех значений всех атрибутов, является своеобразным графическим образом таблицы. Нетрудно заметить, что расположение объектов в таблице влияет на получаемое на битовых картах изображение. Если объекты (в данном случае книги) расположены в строках, то их свойства, имеющие разные типы и форматы и организованные по столбцам, будут отображаться на битовых картах в виде вертикальных линий.

Книжный магазин «Наследие»		Типографский пер., 16		
		еж. 10:00 – 22:00		
Артикул	**Название**	**Автор**	**Год**	**Цена**
Зарубежная проза				
134289	Вино из одуванчиков	Рэй Брэдбери	1957	177 руб.
342565	Холодный дом	Чарльз Диккенс	1853	420 руб.
437903	Принц и нищий	Марк Твен	1881	315 руб.
936756	Айвенго	Вальтер Скотт	1819	261 руб.
673060	Дети капитана Гранта	Жюль Верн	1868	131 руб.
325367	Три мушкетера	Александр Дюма	1844	297 руб.
Русская проза				
056149	Лолита	Набоков В.В.	1955	144 руб.
135934	Спектр	Лукьяненко С.В.	2002	435 руб.
325766	Отцы и дети	Тургенев И.С.	1862	392 руб.
263658	Гроза	Островский А.Н.	1860	130 руб.
786410	Смотритель	Пелевин В.О.	2015	510 руб.
154988	Дубровский	Пушкин А.С.	1841	213 руб.
Поэзия				
123471	Ромео и Джульетта	Уильям Шекспир	1594	388 руб.
388729	Фауст	Иоганн Гете	1960	149 руб.
254869	Евгений Онегин	Пушкин А.С.	1831	355 руб.
209853	Горе от ума	Грибоедов А.С.	1825	267 руб.

1	1	1	1	1
0	0	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
1	1	1	1	1
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
1	1	1	1	1
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	0	0	0	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	0	0	0	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	0	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	0	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0

Рис. 1. Пример составления битовых карт для таблицы

Fig. 1. Bitmaps creation example

В то же время и основные, и врезанные заголовки будут представлять собой горизонтальные линии.

Отметим две важные особенности. Во-первых, врезанные заголовки размещаются горизонтально вне зависимости от ориентации таблицы. Во-вторых, заголовки обычно выделяют из остальных данных с помощью форматирования, а тип данных у них почти всегда строковый, в то время как свойства объектов включают разнообразные типы данных и форматы записи, но редко сильно выделяются форматированием. То есть линии, обнаруженные на картах типов данных и форматов записи, практически однозначно указывают на ориентацию таблицы, а на картах форматирования с большой вероятностью будут обнаружены ни о чем не говорящие горизонтальные линии.

Исходя из описанных особенностей, для определения ориентации таблицы используются только карты типов данных и форматов записи. Преобладание на множестве карт вертикальных линий будет указывать на столбцовую таблицу, горизонтальных – на строчную.

Для обнаружения линий на битовых картах используется метод Хафа [7, 8], в общем случае решающий задачу нахождения на изображении произвольных кривых, заданных параметрическим уравнением. При поиске прямых линий в качестве функции будет выступать уравнение прямой в полярных координатах. Выбор полярных координат обусловлен необходимостью представления вертикальных прямых. В соответствии с методом Хафа составляется матрица для двух параметров уравнения прямой – расстояния до начала координат (R) и угла наклона (θ). Угол наклона искомой прямой в данном случае может принимать значение 0 или 90 градусов, а расстояние R изменяется от 0 до максимального из измерений рабочей области $R_{\max} = \max(R_w, R_h)$ с шагом 1 (рис. 2).

На изображении выбираются точки интереса, для каждой из них ищутся все прямые, проходящие через эту точку, и заполняется матрица Хафа. В качестве точек интереса для конкретной битовой карты рассматриваются клетки, имеющие значение 1 и хотя бы одну клетку со значением 0 по соседству или являющиеся граничными для битовой карты. На рисунке 3 точки интереса закрашены темно-серым цветом.

Заполненная матрица Хафа показывает, на каких прямых лежит больше всего точек, и, соответственно, они наиболее вероятно присутствуют на изображении. Каждая строка итоговой матрицы точек нормируется на максимально возможное значение для данного измерения (то есть длину или ширину таблицы). Затем значения строк объединяются и сортируются. Вычисляется среднее арифме-

R	**0**	**1**	**2**	**3**	**...**	$R_{\max}$
0					...	
90					...	

Рис. 2. Применение метода Хафа для поиска прямых линий

Fig. 2. Using Hough method for detecting straight lines

1	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	0
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1

Рис. 3. Пример выбора точек интереса

Fig. 3. Points of interest selection example

тическое всех значений, и отсекается часть сортированного списка, значения в которой больше среднего арифметического. Если среди оставшихся значений преобладают значения из первой строки матрицы (θ = 0), таблица строчная, в противном случае она столбцовая.

Шаг 3. Поиск шаблона данных и заголовков. Как правило, одному объекту в таблице соответствует одна строка. Однако не в каждой строке таблицы содержится объект. Строка может содержать основной заголовок, врезанный заголовок, агрегатные данные или быть пустой в целях визуального деления таблицы на блоки. Кроме того, в рабочую область попадает не только таблица, но и окружающая ее информация как над, так и под таблицей, которая также не является объектом.

В связи с этим встает задача отделения строк, содержащих объекты, от всех прочих строк. Опираясь на однотипность объектов и их большое количество, данную задачу можно решить статистически, используя битовые карты, созданные на предыдущем шаге алгоритма. В одинаковых столбцах у всех объектов будут содержаться схожие значения, принадлежащие к одному типу данных и, скорее всего, близкие по форматированию. Если рассматривать битовые карты в совокупности, можно обнаружить повторяющиеся шаблоны строк, соответствующие данным и врезанным заголовкам (рис. 4).

При поиске шаблонов целесообразно рассматривать не все имеющиеся битовые карты, во многих случаях можно ограничиться картами типов и форматов данных. Анализируя объединение битовых изображений, выберем все уникальные строки и подсчитаем частоту их повторения. Затем сгруппируем уникальные строки, имеющие не более D отличающихся ячеек, в шаблоны. Шаблоны отличаются от строк тем, что могут содержать ячейки, значения в которых не определены, то есть оно может быть как 0, так и 1. Параметр D в зависимости от количества ячеек в строке (n) принимает следующие значения:

$$D(n) = \begin{cases} 0, \ n < 3, \\ 1, \ 3 < n < 10, \\ [n/5], \ n > 10. \end{cases} \quad (1)$$

Шаблон, которому соответствует наибольшее число строк исходной таблицы, принимается за шаблон данных.

Второй по частоте появления – шаблон врезанных заголовков. Таких шаблонов может быть несколько в случае наличия нескольких уровней заголовков. Все строки, удовлетворяющие шаблону врезанных заголовков, должны располагаться между строками данных. В противном случае они не могут считаться врезанными заголовками. К самой малочисленной группе относятся основные заголовки шапки таблицы, они обычно встречаются в единственном экземпляре, а также строки, не попадающие в вышеприведенные категории. Все выявленные шаблоны используются при построении структурированных объектов по данным, извлекаемым из табличных документов.

Шаг 4. Обработка заголовков и врезанных заголовков. Распознавание заголовков (или шапки) таблицы основано на нескольких предположениях. Во-первых, заголовок таблицы должен находиться непосредственно над первой строкой данных, то есть над первым объектом. Между заголовком и первым объектом может находиться только врезанный заголовок. Во-вторых, заголовок имеет иерархическую (древовидную) структуру. Если заголовок состоит более чем из одной строки и верхние уровни иерархии образованы с помощью объединения ячеек, то у каждой ячейки нижележащего уровня должен быть ровно один родитель на вышележащем уровне. В-третьих, над каждым полем объекта должен находиться ровно один заголовок. Это означает, что заголовок не может содержать пустые ячейки, несколько заголовков не могут соответствовать одному полю и один заголовок нескольким полям.

Учитывая эти предположения, сформулируем последовательность действий при поиске заголовка.

Ищем в исходной таблице первую строку, совпадающую с одним из шаблонов данных.

Выбираем строку, находящуюся над первой строкой данных, и проверяем ее на непротиворечивость описанным выше правилам.

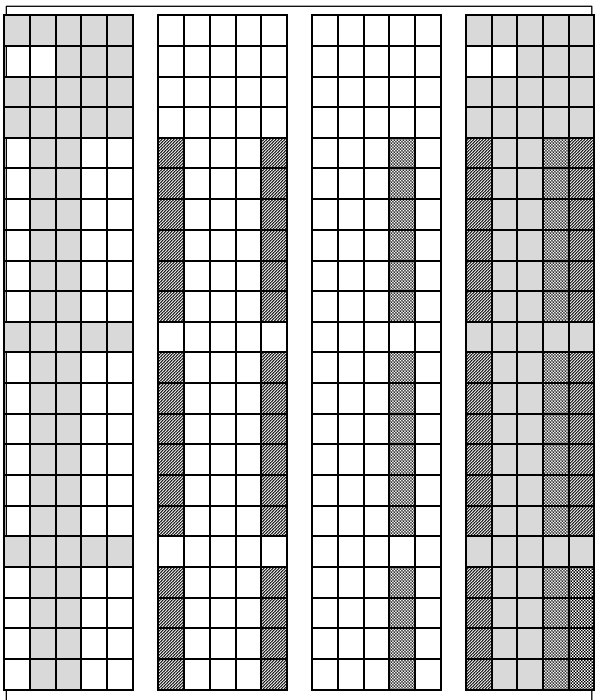

Рис. 4. Поиск шаблона данных по совокупности битовых карт (закрашенные ячейки имеют значение 1)

Fig. 4. Data template search by a set of bitmaps (filled cells have the value of 1)

Если строка противоречит одному из правил, переходим к следующей над ней строке, пока не найдем подходящую.

Когда подходящая строка найдена, она принимается за заголовки нижнего уровня.

Проверяем строку над ней для определения, является ли заголовок многоуровневым. Если вышестоящая строка образует с заголовками нижнего уровня непротиворечивую иерархию, удовлетворяющую описанным выше правилам, она принимается за следующий уровень многоуровневого заголовка. Этот шаг повторяется до тех пор, пока не попадется строка, не вписывающаяся в иерархию. Тогда поиск заголовка прекращается.

В результате выполнения описанных выше шагов получаем размеченную таблицу, каждая строка которой соотнесена с элементом структуры таблицы. Теперь возможно непосредственное извлечение из таблицы полезной информации, то есть содержащихся в ней объектов. Как говорилось ранее, объект представлен набором свойств и их значений, следовательно, из нескольких уровней заголовка нужно сформировать одну строку, которая будет служить именем свойству объекта. Это можно делать, например, так: «Заг.1:Заг.2 ... Заг.N». То есть записать полный путь до заголовка нижнего уровня, используя в качестве разделителя двоеточие. Значения свойств объектов выбираются из строк, соответствующих шаблону данных. Врезанные заголовки можно также рассматривать как свойства со значением true или false, обозначающие принадлежность объекта какой-то категории. Для хранения извлеченных объектов можно использовать формат, подобный JSON (JavaScript Object Notation – описание объекта языка JavaScript).

Тестирование разработанного алгоритма

Тестирование разработанного алгоритма проводилось на наборе таблиц из различных предметных областей. С целью учета всех возможных нюансов распознавания новые документы для тестирования не создавались, а использовались таблицы с реальными данными, составленные разными людьми и находящиеся в свободном доступе. Все использованные таблицы можно разделить на следующие группы:

– простые (одноуровневый заголовок и данные);

– со сложным заголовком (многоуровневые заголовки);

– с врезанными заголовками;

– с многоуровневыми врезанными заголовками.

Оценка эффективности алгоритма распознавания структуры таблицы проводилась отдельно для каждой группы таблиц и типа элементов. Каждая строка может быть распознана как определенный элемент таблицы: данные, заголовок *i*-го уровня, подзаголовок *i*-го уровня и пр. Критерием эффективности выступало отношение правильно распознанных элементов таблицы к общему числу элементов данного типа в конкретной таблице:

$$k = \frac{E_{rec}}{E_{all}} \cdot 100\ \%, \qquad (2)$$

где E_{rec} – количество правильно распознанных элементов; E_{all} – общее количество элементов данного типа; k – коэффициент распознавания.

Общий коэффициент распознавания для всей таблицы определялся по формуле

$$K = \frac{\sum k_i}{N}, \qquad (3)$$

где N – количество различных элементов, $i \in [0, N]$.

Результаты тестирования метода приведены в таблице. Для каждой группы таблиц в первой строке указано отношение количества правильно распознанных элементов заданного типа к общему количеству элементов такого типа в тестовых таблицах. Во второй строке указано количество элементов другого типа, которые были ошибочно распознаны как элементы заданного типа.

Основной заголовок таблицы распознавался методом правильно при условии, что он был корректно составлен и не содержал пустых ячеек. Ошибки в распознавании многоуровневых основных заголовков связаны с тем, что дополнительные данные о таблице, приводимые над ней, в ряде слу-

Результаты тестирования разработанного алгоритма

The results of algorithm testing

Группа таблиц	Уровень основного заголовка			Уровень врезанного заголовка			Данные (объект)	Другое	Распознано таблиц, %
	1-й	2-й	3-й	1-й	2-й	3-й			
Простые таблицы (32 шт.)	30/32	0/0	0/0	0/0	0/0	0/0	12362/12878	41/43	**95**
	0	2	0	493	0	0	1	24	
Многоуровневый заголовок (10 шт.)	10/10	9/10	4/4	0/0	0/0	0/0	3597/3910	8/9	**94**
	0	0	1	281	0	0	0	33	
Врезанные заголовки (23 шт.)	22/23	0/0	0/0	183/197	0/0	0/0	4202/4511	15/17	**92**
	0	0	0	17	306	2	0	1	
Многоуровневые врезанные заголовки (15 шт.)	15/15	0/0	0/0	51/51	85/104	5/12	1637/1742	7/7	**86**
	0	0	0	19	7	97	0	8	
Распознано элементов по типу, %	**96**	**90**	**100**	**94**	**81**	**42**	**95**	**94**	

чаев были оформлены таким образом, что их можно было интерпретировать как следующий уровень иерархии заголовка.

Неточности в обработке врезанных заголовков возникали при отсутствии форматирования, позволяющего отличить один уровень иерархии от другого, например, когда иерархия создавалась за счет многоуровневой нумерации в тексте ячеек. Также негативное влияние на распознавание ячеек данных и врезанных заголовков оказывали частично заполненные таблицы, не все ячейки данных которых содержали значения.

Тем не менее, разработанный метод показал высокую точность распознавания структурных элементов таблицы. Процент корректно распознанных строк для различных видов таблиц составил в среднем 92 %.

Таким образом, в статье рассмотрены современные методы определения структуры таблицы. Предложен метод распознавания структуры таблицы на основании визуальных особенностей отображения, который определяет направление таблицы, шаблоны данных и заголовков. Разработанный метод реализован на языке Java с использованием библиотеки POI для работы с документами форматов XLS и XLSX. Метод показал высокую точность в распознавании структуры таблиц, содержащих большое количество объектов. К таким документам относятся, например, электронные таблицы ценовых предложений. Дальнейшее совершенствование метода возможно за счет использования битовых карт, созданных на основе смысловой близости данных, а также семантического анализа содержимого ячеек.

Работа выполнена в рамках темы НИР № 615869 «Методы проектирования ключевых систем информационной инфраструктуры».

Литература

1. Paine J. Spreadsheet structure discovery with logic programming. Proc. EuSpRIG 2004 Conf. Risk Reduction in End User Computing: Best Practice For Spreadsheet Users In The New Europe, 2004, 11 p.
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back. Proc. 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation, Savannah, GA, USA, 2009.
3. Cunha J., Saraiva J., Visser J. Discovery-based Edit Assistance for Spreadsheets. Proc. 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2009), 2009, pp. 233–237.
4. Oliveira J.N. Transforming data by calculation. Intern. Summer School, Generative and Transformational Techniques in Software Engineering II, Springer, Heidelberg, 2008, vol. 5235, pp. 134–195.
5. Иванов Ю.Н., Емельянов Н.Е., Сотникова Р.А. Документы: типы, описания. Препр. М.: Изд-во ВНИИСИ, 1987. 62 с.
6. Codd E.F. The relational model for database management: vers. 2. Addison-Wesley Longman Publ., 1990.
7. Дегтярева А., Вежневец В. Преобразование Хафа (Hough transform) // Компьютерная графика и мультимедиа. 2003. № 1 (2). URL: http://cgm.computergraphics.ru/content/view/36 (дата обращения: 09.06.2016).
8. Duda R.O., and Hart P.E. Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM 15.1, 1972, pp. 11–15.

Software & Systems Received 10.06.16
DOI: 10.15827/0236-235X.116.107-112 2016, vol. 29, no. 4, pp. 107–112

TABLE STRUCTURE RECOGNITION METHOD FOR SPREADSHEETS DATA

N.M. Tkeshelashvili[1], *Postgraduate Student, nmtkeshelashvili@corp.ifmo.ru*
S.V. Klimenkov[1], *Assistant;* ***A.M. Dergachev***[1], *Ph.D. (Engineering), Associate Professor*

[1] *The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation*

Abstract. Spreadsheets are one of the most popular means to collect and represent big amount of business data. Unfortunately, the structure of spreadsheets in most cases is not defined. As a result, processing tools can not retrieve data automatically, without human interaction. When spreadsheets are being created, people make visual formatting to properly present headers, data and aggregation cells. The proposed method recognizes a spreadsheet structure based on visual elements distinction in a cell formatting as well as people do. The developed software based on this method takes the spreadsheet as an input and produces an output based on a simple object notation form. The method contains several steps. Firstly, a working area is defined as a square area of non-empty cells. After that, the whole working area is converted in the set of bitmaps, where thy value of 1 represents whenever cell has particular formatting. The second step contains discovering data sequence in business data – from the top to the down or from the left to the right. We apply Hough method on a subset of defined bitmaps at this step. Next, we find data and header patterns using simple statistics methods. We define most frequently used patterns as data cells. Special processing is made for subtitles and inserted titles. Such subtitles create additional property to resulting objects. The method has been tested on a large set of spreadsheets containing various business data, for example pricelists.

Keywords: tables processing, structure discover, spreadsheets, visual recognition.

Acknowledgements. *The work has been performed as part of research topic no. 615869 "Design methods for information infrastructure key systems".*

References

1. Paine J. Spreadsheet structure discovery with logic programming. *Proc. of EuSpRIG 2004 Conf. Risk Reduction in End User Computing: Best practice for spreadsheet users in the new Europe*. 2004, 11 p.
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back. *Proc. of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation*. 2009, Savannah, GA, USA.
3. Cunha J., Saraiva J., Visser J. Discovery-based Edit Assistance for Spreadsheets. *Proc. of 2009 IEEE Symp. on Visual Languages and Human-Centric Computing (VL/HCC 2009)*. 2009, pp. 233–237.
4. Oliveira J.N. Transforming data by calculation. *Generative and Transformational Techniques in Software Engineering II.* R. Lammel, J. Saraiva, J. Visser (Eds). International Summer School, vol. 5235, pp. 134–195, Springer Publ., Heidelberg, 2008.
5. Ivanov Yu.N., Emelyanov N.E., Sotnikova R.A. *Dokumenty: tipy, opisaniya* [Documents: Types, Descriptions]. Moscow, VNIISI Publ., 1987, 62 p.
6. Codd E.F. *The relational model for database management: ver. 2*. Addison-Wesley Longman Publ. Co., 1990.
7. Degtyareva A., Vezhnevets V. Preobrazovanie Khafa (Hough Transform). *Kompyuternaya grafika i multimedia* [Computer Graphics and Multimedia]. 2003, iss. 1 (2). Available at: http://cgm.computergraphics.ru/content/view/36 (accessed June 9, 2016).
8. Duda R.O., Hart P.E. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*. 1972, vol. 15.1, pp. 11–15.