Ingenius. Revista de Ciencia y Tecnología
ISSN: 1390-650X
ISSN: 1390-860X
revistaingenius@ups.edu.ec
Universidad Politécnica Salesiana
Ecuador

Algoritmos para el reconocimiento de estructuras de tablas

Escalona Escalona, Yosveni
Algoritmos para el reconocimiento de estructuras de tablas
Ingenius. Revista de Ciencia y Tecnología, núm. 25, 2021
Universidad Politécnica Salesiana, Ecuador
Disponible en: https://www.redalyc.org/articulo.oa?id=505565143005
DOI: https://doi.org/10.17163/ings.n25.2021.05

Algoritmos para el reconocimiento de estructuras de tablas

Algorithms for Table Structure Recognition

Yosveni Escalona Escalona
SOLINTEC, Brasil
yosveni.escalona@solinftec.com.br.
https://orcid.org/0000-0003-2992-0540

DOI: https://doi.org/10.17163/ings.n25.2021.05
Redalyc: https://www.redalyc.org/articulo.oa?id=505565143005

Recepción: 03 Agosto 2020
Revisado: 22 Septiembre 2020
Aprobación: 09 Octubre 2020

Resumen:

Las Tablas son una manera bien común de organizar y publicar datos. Por ejemplo, la Web posee un enorme número de tablas publicadas en HTML integradas en documentos PDF, o que pueden ser simplemente descargadas de páginas Web. Sin embargo, las tablas no siempre son fáciles de interpretar pues poseen una gran variedad de características y son organizadas en diferentes formatos. De hecho, se han desarrollado un gran número de métodos y herramientas para la interpretación de tablas. Este trabajo presenta la implementación de un algoritmo, basado en Campos Aleatorios Condicionales (CRF, Conditional Random Fields), para clasificar las filas de una tabla como fila de encabezado, fila de datos y fila metadatos. La implementación se complementa con dos algoritmos para reconocer tablas en hojas de cálculos, específicamente, basados en reglas y detección de regiones. Finalmente, el trabajo describe los resultados y beneficios obtenidos por la aplicación del algoritmo para tablas HTML, obtenidas desde la Web, y las tablas en forma de hojas de cálculo, descargadas desde el sitio Web de la Agencia Nacional de Petróleo de Brasil.

Palabras clave: Tabular Data, HTML Tables, Spreadsheets, Conditional Random Fields, Machine Learning, Algorithm.

Abstract:

Tables are widely adopted to organize and publish data. For example, the Web has an enormous number of tables, published in HTML, embedded in PDF documents, or that can be simply downloaded from Web pages. However, tables are not always easy to interpret due to the variety of features and formats used. Indeed, a large number of methods and tools have been developed to interpreted tables. This work presents the implementation of an algorithm, based on Conditional Random Fields (CRFs), to classify the rows of a table as header rows, data rows or metadata rows. The implementation is complemented by two algorithms for table recognition in a spreadsheet document, respectively based on rules and on region detection. Finally, the work describes the results and the benefits obtained by applying the implemented algorithm to HTML tables, obtained from the Web, and to spreadsheet tables, downloaded from the Brazilian National Petroleum Agency.

Keywords: datos tabulados, tablas HTML, hoja de cálculo, campos aleatorios condicionales, aprendizaje automático.

Forma sugerida de citación: Excalona Escalona, Y. (2021). «Algoritmos para el reconocimiento de estructuras de tablas». Ingenius. N.° 25, (enero-junio). pp. 50-61. doi: https://doi.org/10.17163/ings.n25.2021.05.

1. Introducción

El volumen de datos disponible en la Internet ha crecido de una manera vertiginosa, lo cual la ha convertido en un vasto repositorio de datos que describen nuestro ambiente y nuestras interacciones. La riqueza y fortaleza de estos datos permiten el desarrollo de la economía y la sociedad hoy en día.

Estos datos están relacionados con información de productos, artículos que imparten conocimiento enciclopédico, presentaciones de resultados científicos de avanzada o reportes sobre datos financieros actuales. Una gran parte de ellos pueden encontrarse en tablas, que requieren un análisis particular ya que pueden estar expresadas en HTML, integradas en documentos PDF o estar disponibles como hojas de cálculo

descargables, entre otros formatos. Usualmente, las tablas se organizan de forma simple y compacta como filas y columnas, pero pueden ser más complejas con metadatos e información adicional.

Las tablas han demostrado ser fuentes valiosas, pero su uso puede estar muy diversificado, desde la búsqueda en la web hasta el descubrimiento de datos en hojas de cálculo y aumento de bases de conocimiento [1]. En la literatura se encuentran estudios sobre métodos y herramientas para la extracción de datos tabulares de hojas de cálculo, tablas HTML, tablas integradas en documentos PDF, etc. La gran mayoría de estos métodos y herramientas utilizan estrategias basadas en reglas heurísticas y algoritmos de aprendizaje de máquina. La estrategia para extraer datos tabulares y para clasificar filas de tablas también depende del formato del documento. Explorar un conjunto grande de tablas ha sido un reto porque, en general, la semántica de la tabla es desconocida. En [2], se presenta un corpus de más de cien millones de tablas, pero el significado de cada tabla raramente está explícito en la misma tabla. Otro reto es la estructura de la tabla. Por ejemplo, las tareas descritas en [3], [4], [5], [6] se enfocan en recuperar la semántica de la tabla y en vincular sus datos con fuentes externas para tablas clasificadas como genuinas, con una pérdida considerable de datos. Estos trabajos no consideran aspectos fundamentales, tales como la orientación de la tabla y descartan aquellas tablas clasificadas como no genuinas.

Otro aspecto a considerar está basado en el tipo de documento, por ejemplo, Correa y Zander [7] analizaron un grupo de métodos y herramientas enfocados en extraer contenido tabular de archivos PDF basándose en dos características principales: facilidad de uso y resultados de salida y la categorización de las herramientas según propuestas teóricas, sin costo y comerciales. En [8] se desarrollaron varias heurísticas, que conjuntamente reconocen y descomponen tablas en archivos PDF y almacenan los datos extraídos en un formato estructurado de datos (XML) para facilitar su uso, estas heurísticas se dividen en dos grupos: reconocimiento y descomposición de tablas. Otras técnicas fueron presentadas en [9] para extraer data tabular de documentos PDF con el fin de identificar los límites de la tabla, donde los autores describen una metodología que aplica dos algoritmos de aprendizaje de máquina, CRF y máquinas de soporte vectorial (SVM, Support Vector Machines). Asimismo, se han revisado trabajos basados en el proceso de identificación de límites de tabla y diseñados para la correspondencia semántica y anotación de atributos numéricos y variantes en el tiempo en tablas web como las presentadas en [10], [11],[12] que anotan tablas web efectiva y eficientemente, e identifican los límites entre filas (o columnas) de nombres de atributos y sus correspondientes filas (o columnas) de valores en la tabla.

También se puede hacer mención especial de los trabajos relacionados con el reconocimiento de la estructura de una tabla HTML y la detección y clasificación del encabezado de una tabla descritas en [13], [14], sugiriendo algunas técnicas basadas en reglas heurísticas que utilizaron un algoritmo de aprendizaje de clasificación para delinear tipos de tablas existentes dentro de un documento y detectar los tipos de estructuras y encabezados.

Finalmente, se hace énfasis en el enfoque propuesto en [15] que estuvo basado en técnicas de aprendizaje de máquina que cubren dos tareas fundamentales del proceso de extracción de una tabla: su localización e identificación de las posiciones y tipos de filas. Este trabajo se enfoca en la implementación de dos algoritmos para el reconocimiento de tablas en hojas de cálculo, así como también otros algoritmos basados en campos aleatorios condicionales (CRF), para clasificar los tipos de filas dentro de las tablas. Los conjuntos de datos fueron creados con tablas HTML descargadas; las tablas de hojas de cálculo fueron obtenidas del sitio web de la Asociación Nacional de Petróleo (ANP) de Brasil.

1.1 Contexto

Las tablas se encuentran frecuentemente en documentos impresos, libros o periódicos, así como también en documentos digitales, páginas electrónicas o láminas de presentación. Sin embargo, dichas tablas también representan un concepto esencial en bases de datos relacionales y hojas de cálculo. Las tablas pueden

distinguirse de acuerdo con su estructura y orientación. Una tabla relacional u horizontal [8], como la que se ilustra en la Tabla 1, tiene filas que proporcionan datos sobre objetos específicos llamados entidades y columnas, que representan atributos que describen las entidades.

ID	Nombre	Edad	País	Ocupación
1	Bob Smith	35	USA	Programador
2	Jane Smith	31	USA	Profesora
3	Robert White	24	UK	Ingeniero

Tabla 1. Ejemplo de una tabla relacional

Existen tablas más complejas, como aquellas donde los atributos que describen las entidades están colocados verticalmente y las entidades de manera horizontal, u otro tipo de estructuras como las que se muestran en la Tabla 2 y en la Tabla 3.

	Obj 1	Obj 2	Obj 3
Nombre	V1	V1	V1
Edad	V2	V2	V2
Altura	V3	V3	V3

Tabla 2. Ejemplo de una tabla no relacional

Aplicaciones de Patentes por Residentes		
Fuente de datos: worldbanc.org (se muestra país tope en cada continente)		
País	**Residentes**	**Aplicaciones**
Norte América		
Estados Unidos	307,700,000	224,912
Canadá	33,739,900	5,067
Asia		
Japón	127,557,958	295,315
China	1,331,380,000	229,096

Tabla 3. Ejemplo de una tabla no relacional con información adicional

De manera más precisa, una tabla se define como:

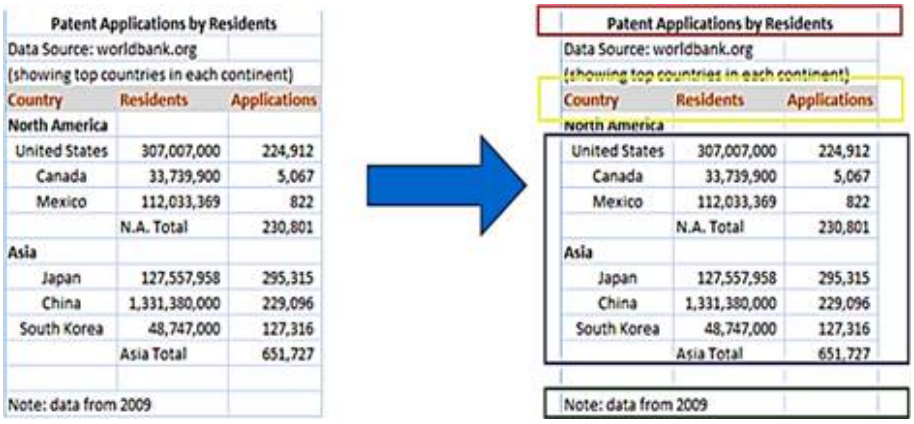
Definición 1: Una tabla es un par T = (H, D) que consiste de un encabezado opcional H y datos D, donde:

• El encabezado H = {h1, h2, . . . , hn} es una n-tupla de elementos hi de encabezado; si el conjunto de elementos de encabezado existe, este podría ser representado como una fila o como una columna.

• Los nodos de datos están organizados como una matriz (n, m) que consiste de n filas y m columnas:

$$D = \begin{bmatrix} C_{11} & \therefore \cdots & C_{1m} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nm} \end{bmatrix}$$

El proceso de clasificación de las filas de una tabla consiste en identificar cada uno de los elementos de una tabla. La idea general está basada en localizar el encabezado y los datos en la tabla. También es relevante identificar los elementos de diseño y los metadatos. La Figura 1 muestra el proceso de clasificación de las filas de una tabla, denotando en diferentes colores algunos de los elementos presentes en la tabla: rojo indica los elementos que representan los títulos; amarillo, la fila de encabezado; azul, los datos de las filas; y verde, los metadatos adicionales.



Patent Applications by Residents		
Data Source: worldbank.org		
(showing top countries in each continent)		
Country	Residents	Applications
North America		
United States	307,007,000	224,912
Canada	33,739,900	5,067
Mexico	112,033,369	822
	N.A. Total	230,801
Asia		
Japan	127,557,958	295,315
China	1,331,380,000	229,096
South Korea	48,747,000	127,316
	Asia Total	651,727
Note: data from 2009		

Figura 1. Proceso de extracción tabular

El resto del artículo está organizado como sigue. La sección 2 cubre detalles de las implementaciones de algoritmos para el reconocimiento de tablas y para la clasificación de filas de tablas. Finalmente, la sección 3 describe experimentos y resultados.

2. Materiales y métodos

2.1. Algoritmos de reconocimiento de tablas y de clasificación de filas de tablas

Esta sección describe las implementaciones de un algoritmo para reconocer tablas en hojas de cálculos y un algoritmo, basado en campos aleatorios condicionales, para clasificar filas de tablas.

2.1.1. Un algoritmo basado en reglas para detectar tablas en hojas de cálculo

Varios algoritmos basados en reglas detectan tablas en hojas de cálculo utilizando atributos de celdas, tales como bordes, formato y tipo de dato. El atributo de cada celda en la hoja de cálculo tiene un valor específico asociado con esa celda. A su vez, el borde de la celda tiene los atributos de dirección, estilo y color. El borde puede rodear la celda en 4 direcciones diferentes: arriba, abajo, izquierda y derecha.

Un formato de celda es el formateo visual aplicado al dato de la celda, tal como, formato de número, nombre de enegrita, letra cursiva y color de letra. La detección de múltiples tablas en la misma hoja de cálculo se realiza encontrando un separador entre dos tablas (usualmente un conjunto de filas vacías), como se explica a continuación [16]. stilo de letra, nombre de letra, tamaño de letra, letra

Dada una tabla T, con rn filas y cn columnas, se calculan las siguientes características de diseño:

• Número promedio de columnas, calculado como el número promedio de celdas por fila.

$$c = \frac{1}{rn} \sum_{i=1}^{rn} c_i \tag{1}$$

donde ci es el número de celdas en la fila i, i=1, . . . rn.

• Número promedio de filas, calculado como el número promedio de celdas por columna.

$$r = \frac{1}{cn} \sum_{i=1}^{cn} r_i \tag{2}$$

donde ri es el número de celdas en la columna i, i = 1, . . . , cn.

La Figura 2 muestra el algoritmo que identifica el número de tablas dentro de un documento y captura el rango de filas que representa cada tabla.

Algorithm 1 Table Detection and Recognition

Input: A spreadsheet document
Output: Number of tables into the document (ct)

$U \leftarrow$ The threshold to empty rows
$H \leftarrow$ Set header cells
$D \leftarrow$ Set data cells
$T \leftarrow$ Set title cells
$E \leftarrow$ Set empty rows
$R_i \leftarrow$ Numbers of cells tagged as header in row i
$X \leftarrow$ Numbers of Empty cells between a header row and title row
for $i \leftarrow 0$ to rn **do** ▷ Number of rows into the spreadsheet document
  for $j \leftarrow 1$ to cn **do** ▷ Number of columns into the spreadsheet
    if $C[i-1,j]$ **is** $Header$ **and** $C[i-1,j].type = String$ **then**
      if $C[i,j].format = C[i,j-1].format = C[i,j+1].format$ **and** $C[i-1,j].format = C[i-1,j].format = C[i,j].font = C[i-1,j].font$ **then**
        if $R_i > 0$ **then**
          $C[i,j] \in Header$
          if $C[i-1,j] \neq Empty$ **and** $C[i,j].format \neq C[i,j].format$ **then**
            $H \leftarrow H \cup \{c\}$ ▷ Adds c to the set of header cells
          end if
        end if
      end if
    else if $C[i,j]$ **and** $C[i-1,j]$ **and** $C[i+1,j] \in \{type, format\}$ **then**
      $D \leftarrow D \cup \{c\}$ ▷ Adds c to the set of data cells
    else if $C[i,j].type = String$ **and** $E \geq X$ **then**
      if $C[i,j-1]$ **and** $C[i,j+1] \in E$ **and** j *is first column* **then**
        $T \leftarrow T \cup \{c\}$ ▷ Adds c to the set of title cells
      else
        $E \leftarrow E \cup \{c\}$ ▷ Adds c to the set of empty cells
      end if
    end if
  end for
  if $len(E) = U$ **then** ▷ if the count of empty cells equals to threshold
    $ct \leftarrow ct + 1$
  end if
end for
return ct

Figura 2. Algoritmo de detección y reconocimiento de tabla

Detección de regiones

La detección de regiones se calcula a través de un algoritmo basado en grafos denominado Remover y conquistar [17], que detecta tablas en hojas de cálculo. Este algoritmo utiliza un conjunto completo de reglas y heurísticas de acuerdo con una representación de una hoja de cálculo como un grafo. Los archivos de hojas de cálculo contienen una o más hojas, cada hoja consta de una colección de celdas organizadas en filas y columnas, donde se definen ciertos términos útiles para el proceso de detección de regiones

Definición 2. Sea W el conjunto que contiene todas las celdas de una hoja.

La detección de una región consiste en escanear la hoja de cálculo desde la primera celda en la esquina superior izquierda hasta la última celda no vacía en la esquina inferior derecha, para chequear celdas con formato similar y detectar separadores, tales como filas vacías, diferentes formatos de celdas o diferentes tipos de bordes, tales como diferentes tipos de valores de celdas. De manera más precisa, una región se define como:

Definición 3. Una región es una colección máxima R # W de celdas de un área rectangular de la hoja.

También se infiere el rol de diseño de las celdas no vacías en la hoja, donde a cada celda no vacía se le asignan los siguientes roles: Encabezado (H, Header), Datos (D), Título (T), Metadato o no relacional (N). Este rol de celda se define como sigue.

Definición 4. Sea la label: W → Etiquetas, donde Etiquetas = {Encabezado, Datos, Título, Metadatos}, una función que relaciona a las celdas su rol de diseño asignado.

Para celdas vacías, la etiqueta no está definida; estas celdas se identifican utilizando empty: W → {0, 1}, que retorna 1 para celdas vacías y 0 en otro caso. Las celdas de una hoja de cálculo se agrupan conjuntamente, de manera que celdas adyacentes tengan el mismo rol de diseño (etiqueta) o formen estructuras más grandes. Estos grupos se denominan regiones etiqueta, como se muestra en [17] y en la Figura 3.

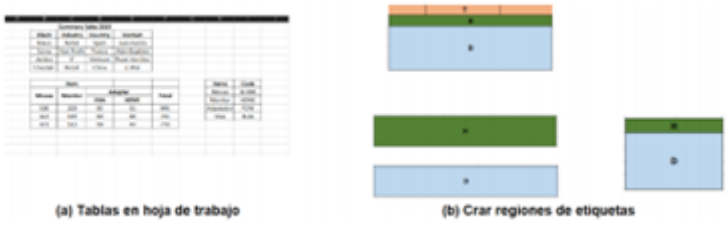


Figura 3. Proceso de creación de regiones etiquetas

Formalmente, una región etiqueta se define de la siguiente manera:

Definición 5. Una región etiqueta es una región LR de una hoja de cálculo tal que, para cualesquiera dos celdas c y c 0 en LR, label(c) = label(c 0) y empty(c) 6= 1 y empty(c 0) 6= 1.

La Figura 4(a) muestra tablas en una hoja de cálculo y la Figura 4(b) indica las regiones correspondientes a las estructuras de la tabla. El proceso de detección de regiones etiqueta agrupa celdas según su etiqueta. Se itera a lo largo de cada fila para crear secuencias de celdas que tienen la misma etiqueta. Estas forman la base de LR. Posteriormente, se unen las LR de filas consecutivas, si sus etiquetas, columna mínima y columna máxima coinciden.

Representación de tablas mediante grafos

Las regiones permiten construir grafos que capturan las interrelaciones de regiones etiqueta. La Figura 4 muestra la representación de tablas como un grafo.

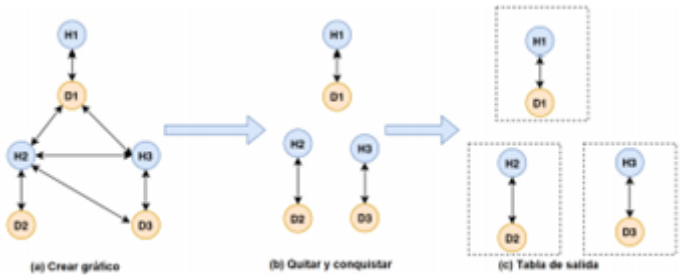


Figura 4. Representación de tablas mediante grafos

El proceso de construcción de grafos consiste en identificar relaciones espaciales: superior, inferior, izquierda y derecha, localizando las regiones vecinas más cercanas para cada dirección, e identificar todos los vértices cuya máxima fila es menor que la mínima fila de otro vértice. Para cada dirección, se define una función distancia donde se identifican todos los vértices más cercanos:

$$ND_v = \{n \in D_v \vee ddist(v,n) = \underset{u \in DV}{minddist}(u,v)\}$$

(3)

donde Dv es la dirección para el vértice v; bordes dirigidos (v, n) se crean para cada n ∈ NDv.

2.1.2. Algoritmo Remover y conquistar

Remover y conquistar (RAC, Remove and Conquer) es un algoritmo basado en reglas cuyo objetivo es separar los bordes que están más lejos hacia las direcciones izquierda y derecha del grafo que fue creado a partir de cada hoja en una hoja de cálculo, como se muestra en la Figura 5. El algoritmo procesa los componentes fuertemente conectados del grafo, para aparear todos los grupos formados y detectar tablas válidas.

Los vértices se ordenan en orden descendente de su máxima fila, seguido por orden ascendente de su mínima fila, por lo que las tablas se buscan en orden inverso, desde abajo hacia arriba. Cada encabezado h se procesa individualmente para identificar vértices con fila mínima mayor o igual que h.

El algoritmo que verifica el encabezado válido se muestra en la Figura 6. Todos los encabezados válidos son almacenados en Q que representa el conjunto de vértices, incluyendo h; este conjunto de vértices se denomina tablas potenciales. El algoritmo asegura que otros vértices conectados a h no se dejen aislados.

Algorithm 2 Remove and Conquer

Input: Graph representation of a worksheet
Output: Tables into document

$P \leftarrow 0$
$E_l \leftarrow \{e \in E / dir(e) = Left \text{ and } ldist(e) > 1\}$
$E_r \leftarrow \{e \in E / dir(e) = Right \text{ and } rdist(e) > 1\}$
$E \leftarrow (E_l \cup E_r)$
for $G^* \in getSCC(G)$ **do**
  $LQ \leftarrow Null$
  $S^{'} \leftarrow \{v \leftarrow S^{'} / lbl(v) = Header\}$
  if $|S^{'}_H| > 0$ **then**
    for $h \leftarrow S^{'}_H$ **do**
      if $h \leftarrow LQ$ **then**
        $Q \leftarrow \{s \in S / rmin(s) \geq rmin(h) \text{ and } hasPath(s, h, E_s)\}$
      end if
      if $isValid(h, Q, 0.5)$ **then**
        $P \leftarrow P \cup \{LQ\}$
        $S^{'} \leftarrow S^{'} / Q$
      else if $LQ = Null$ **then**
        if $|Q| = 1$ **and** $isAligned(h, LQ)$ **then**
          $LQ \leftarrow LQ \cup \{h\}$
        end if
      end if
    end for
  end if
  $P \leftarrow P \cup \{LQ\}$
end for
$U \leftarrow U \cup \{S\}$ ▷ Remaining unpaired
$P, U \leftarrow handleOverlapping(P, U)$
for $u \in U$ **do** ▷ Find nearest table left or right
  $N, dist \leftarrow getNearestVertices(u, (E_l \cup E_r))$
  $P^{'} \leftarrow \{P \leftarrow P / 0 < |N \cap P|\}$
  if $|P^{'}| = 1$ **and** $dist \leq 3$ **then**
    $P \leftarrow P \cup \{u\}, where\ P \in P^{'}$
  end if
end for
return P, U

Figura 5. Algoritmo Remover y Conquistar

Input: h: a Header vertex, Q: vertices to form table with, th: threshold for alignment ratio
Output: $True$ if h is valid, $False$ otherwise
1: **if** $|\{q \in Q | rmin(q) > rmax(h)\}| > 0$ **then**
2: $Q_{\mathcal{H}} \leftarrow \{q \in Q | lbl(q) = Header$ **and** $rmin(q) \leq rmax(h)$ **and** $rmin(q) \geq rmin(h)\}$
3: $X \leftarrow \emptyset$; $X' \leftarrow \emptyset$
4: **for all** $u \in Q_{\mathcal{H}}$ **do**
5: $X \leftarrow X \cup \{x \in \mathbb{N} | cmin(u) \leq x \leq cmax(u)\}$
6: **for all** $v \in Q \setminus Q_{\mathcal{H}}$ **do**
7: $X' \leftarrow X' \cup \{x \in \mathbb{N} | cmin(v) \leq x \leq cmax(v)\}$
8: **return** $\frac{|X \cap X'|}{|X'|} \geq th$ **and** $|X| > 1$
9: **else**
10: **return** $False$

Figura 6. Chequeo de validez del encabezado

Esos vértices apareados con un encabezado válido son sustraídos del conjunto de vértices y luego ordenados para crear el conjunto S'. Los encabezados válidos son agregados al conjunto de encabezados válidos, llamado LQ. Los vértices que representan tablas potenciales, llamados Q, no son directamente agregados al conjunto de tablas P porque el algoritmo necesita chequear que h no está conectado a otros vértices. Las tablas que no pueden formarse se almacenan en U. Entonces, en el último paso del algoritmo, este intenta aparear las tablas en U con la tabla más cercana a su izquierda o derecha.

2.1.3. Un algoritmo de aprendizaje de máquina para clasificar filas de tabla

Una contribución importante de este trabajo es la identificación y clasificación de los tipos de filas que componen una tabla, a través de la implementación de un algoritmo de aprendizaje de máquina, en este caso, campos aleatorios condicionales (CRF), el cual está basado en las características, valores de las celdas, así como también las clases que representan la estructura de la tabla.

Los CRF son modelos de grafos que no tienen dirección, introducidos por Lafferty et al. [18], que pueden actuar como clasificadores en tareas de etiquetado de secuencias. Estos son utilizados frecuentemente para procesamiento de lenguaje natural, tal como etiquetado de partes de discursos. El algoritmo CRF define X como una variable aleatoria sobre las secuencias de datos a ser etiquetadas y Y como una variable sobre las secuencias de etiquetas correspondientes. La Figura 7 muestra una estructura de un campo aleatorio condicional lineal.

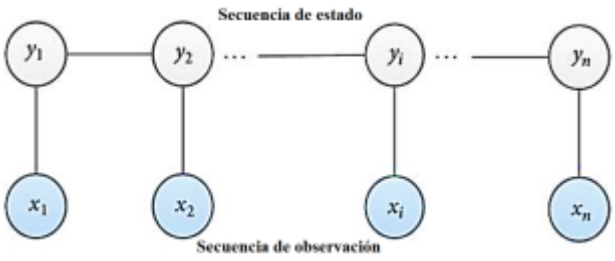


Figura 7. Estructura de un campo aleatorio condicional de cadena lineal

En nuestro problema de clasificar filas de tablas, la secuencia de entrada x corresponde a una serie de filas de una tabla dada, mientras que la secuencia de etiquetas y es la serie de etiquetas asignadas a las filas observadas. A cada fila en x se le asigna exactamente una etiqueta en y.

Formalmente, los campos condicionales aleatorios se definen de la siguiente manera:

Definición 6. Sea G = (V, E) un grafo y Y = $(Y_V)_{V \in V}$ una secuencia de variables aleatorias indexadas por los vértices de G. Un campo aleatorio condicional es un par (X, Y) tal que, cuando se condiciona en X, las variables aleatorias Yv obedecen la propiedad de Markov con respecto al grafo.

$$P(Y_v|X, Y_w, w \neq v = P(Y_v \vee X, Y_w \approx v) \tag{4}$$

$$P(X \vee Y) = \frac{1}{Z(x)} exp(\sum_j \lambda_j f_j(Y_{i-1}, Y_i, X, i) + \\ + \sum_k \mu_k g_k(Y_i, X, i)) \tag{5}$$

donde fj (Yi−1, Yi , X, i) es una función de características de transición de la secuencia de observaciones y de las etiquetas en las posiciones i e i−1 en la secuencia de etiquetas; gk(Yi , X, i) es una función de características de estado de la etiqueta en la posición i y la secuencia de observaciones; y λj y μk son parámetros a ser estimados a partir de datos de entrenamiento.

En el escenario de la tabla de datos, X representa la lista de filas en la tabla y Y representa las clases de fila correspondientes. Cada tabla de datos relacional tiene un esquema, el cual, en el contexto de las tablas de datos, consiste de nombres de atributos, valores y tipos, donde los nombres de atributos son títulos de columnas, los tipos de atributos son los tipos de valores en la columna y los atributos de valor corresponden a valores de datos en las celdas de las columnas. Los nombres de las columnas son almacenados en una fila o en filas especiales, usualmente cerca del encabezado de la tabla, llamadas filas de encabezado, mientras que los datos son almacenados en filas referidas como filas de datos.

La tabla de datos también puede contener descripciones de datos referidas a los metadatos. En correspondencia con los criterios tratados arriba, se identifica cada tipo de fila de acuerdo con las propiedades de cada celda en la tabla de datos. Entonces, el problema se enfoca en asignar una etiqueta a cada fila, donde cada fila está constituida por celdas que pueden exhibir diferentes conjuntos de atributos. El proceso de selección de características involucra la extracción de una colección de atributos para celdas individuales y combinar los atributos de todas las celdas en la fila, con el fin de construir un conjunto de características de filas. Considere las ideas abordadas arriba y un ejemplo de una tabla simple con encabezado y datos, tal como se muestra en la Figura 8.

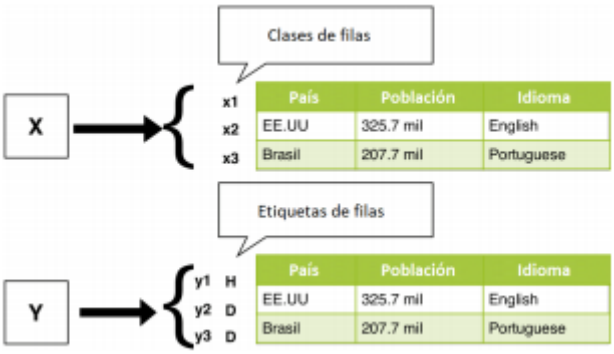


Figura 8. Ejemplo de una tabla etiquetada

X representa un vector con las filas de la tabla y Y representa otro vector con las etiquetas de cada fila x de la tabla.

2.1.4. Clases de filas

De acuerdo con la estructura de tablas y la Definición 6, se definieron los tipos de clases de filas que se muestran en la Tabla 4.

Etiqueta	Descripción
H	Representa la fila de encabezado en la tabla
D	Filas de datos que contienen registros de datos
N	Metadatos no relacionales

Tabla 4. Clases de filas

2.1.5. Conjunto de características

En cualquier algoritmo de aprendizaje de máquina, una característica es una propiedad individual medible o una característica de un fenómeno que está siendo observado [19]. Por lo tanto, cada característica se particiona en tres categorías considerando aspectos relacionados al diseño, estilos y valores que pueden llamarse atributos de diseño.

Atributos de diseño son las celdas que se encuentran comúnmente en filas de encabezado, que usualmente contienen celdas de tablas combinadas con texto centrado.

Atributos de estilo son varias propiedades derivadas de hojas de estilo, tales como tipo de letra, color de letra, peso de letra o texto subrayado.

Atributos de valor son aquellos que representan celdas donde la información almacenada está vinculada exclusivamente con las filas de datos. Frecuentemente, las filas de encabezado contienen valores textuales relativamente cortos, en lugar de números o fechas.

Observe con un ejemplo como trabaja el algoritmo CRF en nuestro problema de clasificación de tabla, dada la característica de transición fj (Yi−1, Yi , X, i) y la función de características gk(Yi , X, i):

x es una fila dentro de la tabla de datos.

j is la posición de una fila en la tabla (cada característica está asociada con una posición); más de una característica asociada con la misma posición.

yj−1, yj son las etiquetas (clases) asignadas a las filas j y j − 1 de x

Entonces, la función de características y la función de estados son las siguientes (6), (7), (8), (9):

$$f_1(Y_{i-1}, Y_i, X, i) = \begin{cases} 1 if x_j \in heady_j = H \\ 0, otherwise \end{cases} \quad (6)$$

$$f_2(Y_{i-1}, Y_i, X, i) = \begin{cases} 1 if x_j \in daty_j = D \\ 0, otherwise \end{cases} \quad (7)$$

$$g_1(Y_i, X, i) = \begin{cases} 1 if (x_j isacell \in x) \wedge (x_j \in rowfeatures) \wedge y_j \\ 0, otherwise \end{cases} \quad (8)$$

$$g_2(Y_i, X, i) = \begin{cases} 1 if (x_j isacell \in x) \wedge (x_j \in rowfeatures) \wedge y_j = D \\ 0, otherwise \end{cases} \quad (9)$$

La lista completa de atributos de celdas individuales está dada en la Tabla 5. Las características se dividen de según el tipo de atributos que ellas representan.

Diseño	Estilo	Valor	Espacial
IsMerged	IsBold	IsEmpty	RowNumber
Aligment	IsItalic	IsText	ColNumber
	IsUnderlined	IsNumber	NumNeighbor
	IsColored	IsDate	MatchStyle
	Font	IsAlpNum	MatchType
	Format	IsCapital	
	Border	TotalWord	

Tabla 5. Atributos de celdas

2.1.6. Similitud entre filas

Otra característica que fue tomada en cuenta para generalizar los datos de entrenamiento fue la similitud entre filas [20], donde se asigna una característica única a cada combinación única (c, r) donde c es el número de celdas que exhiben un atributo y r es el número de celdas en la fila. Entonces, dos filas Rx y Ry se consideran similares con respecto a cierto atributo de celda α si el logaritmo de sus anchuras son iguales y el logaritmo del número de celdas que exhiben o carecen del atributo α. Este enfoque se conoce como agrupación de características y puede definirse como sigue.

Formalmente, para una fila Ri de longitud r en la cual c celdas exhiben un atributo específico de celda α, se la asigna la característica "Rα = (a, b)" a Rj (a, b), donde a y b son los grupos que se calculan como sigue (10), (11).

$$a = \begin{cases} 0, & if c = 0 \\ \lfloor log_2(c) + 1 \rfloor, & if 0 < c \leq r/2 \\ \lfloor log_2(r - c) + 1 \rfloor^-, & if\ r/2 < c < r \\ 0^- & if c = r \end{cases} \quad (10)$$

$$b = \lfloor log_2(r) \rfloor \quad (11)$$

Los objetivos de los grupos son:

1. Diferenciar entre anchos de tablas
2. Combinar tablas anchas
3. Resaltar filas uniformes

3. Resultados y discusión

Esta sección presenta los experimentos realizados para probar la exactitud de la implementación del clasificador de filas de tablas, así como también los experimentos con reconocimientos de tablas en documentos con hojas de cálculos.

3.1. Preprocesamiento

La tarea de preprocesamiento se enfoca en dos escenarios de tablas: tablas HTML y tablas de hojas de cálculo, con el fin de remover contenido irrelevante o contenido que no proporcionará información para el clasificador de filas de tablas. Otros aspectos que fueron considerados fueron las estructuras de las tablas y la información presente en ambos tipos de tablas. Este trabajo no cubre en completo detalle el preprocesamiento de las tablas, por lo que solo se hace énfasis en aquellos elementos que se consideran más importantes. En el caso de tablas de hojas de cálculo, se resalta que el conjunto de datos tenía una anotación predefinida, pero con muchos errores relacionados con la identificación de los rangos de filas de datos y filas de encabezado.

3.2. Características principales de los conjuntos de datos utilizados para prueba

La Tabla 6 muestra las estadísticas del proceso de anotación en ambos conjuntos de datos. Cada fila de cada tabla fue anotada con la etiqueta correspondiente a su clase: «H» para encabezamiento (Header), «D» para datos, etc.

	HTML	Hoja de cálculo
Tablas anotadas	105	252
Filas anotadas	13,025	227,638
Filas de encabezado	105(<1%)	252(<1%)
Filas de datos	12,920(99%)	227,254(98%)
Otras clases de filas	0(0%)	132(<1%)

Tabla 6. Tablas anotadas

La tabla arriba indica que un aspecto crítico de tablas tanto HTML como hojas de cálculo es que el porcentaje de filas de encabezado es muy bajo, debido al hecho de que las tablas obtenidas fueron tablas simples con esquemas simples (tablas con una sola fila de encabezado seguida por una o más filas de datos).

3.3. Experimentos de clasificación de tablas

Esta sección presenta los experimentos para evaluar la solución propuesta de clasificación de tablas. En una primera etapa, se entrenó el algoritmo con 80 % de los datos y se probó con 20 % de los datos, seleccionados aleatoriamente. Se utilizó el algoritmo L-BFGS como método de optimización y parámetros de regularización L1 y L2 ajustados a 0.1 y 0.01. Los experimentos con tablas HTML y hojas de cálculo fueron realizados de manera separada, para exponer las diferencias entre los dos formatos de tabla. Las métricas de desempeño adoptadas fueron precisión, memoria, f1-score, soporte.

3.3.1. Resultados

Esta sección muestra los resultados obtenidos. Se observa que el valor de precisión para tablas de hojas de cálculo fue mayor que para tablas HTML, debido a dos factores principales: (1) las características de las tablas de hojas de cálculo tienen una mejor definición; (2) se garantiza una correcta definición para las filas de datos. La memoria fue similar para ambos tipos de tablas, así como también el f1-score. Un punto importante en este análisis está relacionado con el número de filas clasificadas como no relacionales en el conjunto de datos de hojas de cálculo, debido al hecho de que se anotaron manualmente las tablas de hojas de cálculo, a diferencia de las HTML, donde algunas filas podrían haber sido identificadas como filas de datos o filas de encabezado, siendo de hecho filas no relacionales (Tabla 7).

Clase de fila	Precisión	Memoria	F1-Score	Soporte
HTML				
D	0.966	0.982	0.970	2,496
H	0.955	0.992	0.970	17
N	0.980	0.980	0.970	92
Hojas de cálculo				
D	0.997	0.985	0.994	39,08
H	0.969	0.993	0.983	49
N	0.985	0.965	0.974	5

Tabla 7. Resultados para tablas HTML y hojas de cálculo

Nota: las etiquetas de fila son como en la Tabla 4:

D: Filas de datos
H: Filas de encabezado
N: Metadatos no relacionales (una nota, clarificación, etc.)

3.3.2. Validación cruzada

Validación es el proceso de decidir si los resultados numéricos que cuantifican las hipótesis entre variables son aceptables como descripciones de los datos; este un proceso útil cuando no existen datos suficientes para entrenar el modelo y existe una gran desequilibrio en el número de objetos en cada clase. Entonces, se aplicó

una estrategia k-fold conocida como k-fold estratificado, que es una ligera variación de la estrategia k-fold de validación cruzada, tal que el fold contiene aproximadamente el mismo porcentaje de muestras de cada clase objetivo que el conjunto completo.

La Tabla 8 muestra los resultados obtenidos para ambos conjuntos de datos. Se observa que para el caso de tablas HTML, los mejores resultados se obtuvieron para k = 2 y k = 3 y que la precisión promedio fue 0,958 y que para las tablas de hojas de cálculo cada k = 1 ..., 5 es similar y la precisión promedio fue 0,997.

HTML					
Etapa	**K = 1**	**K = 2**	**K = 3**	**K = 4**	**K = 5**
Prueba	0,92	0,98	0,98	0,94	0,97
Hojas de cálculo					
Etapa	**K = 1**	**K = 2**	**K = 3**	**K = 4**	**K = 5**
Prueba	0,997	0,998	0,996	0,998	0,996

Tabla 8. Precisión del método de validación cruzada para tablas HTML y hojas de cálculo

3.3.3. Matriz de confusión

Como en cualquier problema de clasificación, existen aspectos que pueden ser mejorados. En nuestros experimentos, se tienen que examinar las filas en cada clase que fueron confundidas con filas en otra clase. Entonces, se utilizó una matriz de confusión, como se muestran en la Figura 9 y en la Figura 10. Cada celda de la matriz muestra el porcentaje de todas las filas clasificadas que fueron realmente de la clase con la etiqueta mostrada en la primera columna, pero que el clasificador le asignó la etiqueta de fila mostrada en la primera fila. Las celdas sombreadas con color azul más oscuro en la diagonal muestran clasificaciones correctas de filas, mientras que las remanentes muestran clasificaciones incorrectas.

Idealmente, nuestro clasificador resultaría en ceros para los valores fuera de la diagonal. Sin embargo, de hecho, el modelo clasificó filas incorrectamente. En el caso de tablas de hojas de cálculo, se observó que, tanto para filas de datos como para filas de encabezado, se obtuvieron resultados erróneos con respecto a las filas no relacionales, esto es, un número considerable de filas de datos y filas de encabezado fueron identificadas como filas no relacionales. En las tablas HTML, los resultados erróneos para filas no relacionales fueron mayores que para las tablas de hojas de cálculo, siendo 7,9 % y 6,6 % para filas de datos y filas de encabezado, respectivamente.

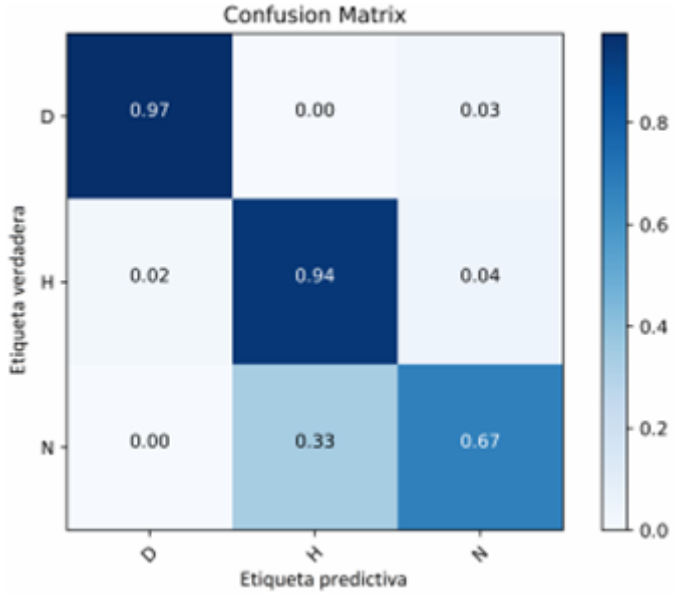


Figura 9. Matriz de confusión para tablas de hojas de cálculo

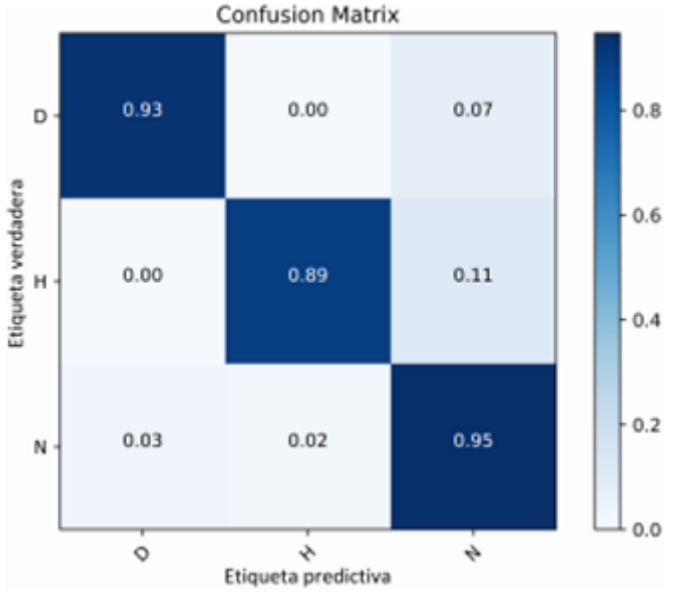


Figura 10. Matriz de confusión para tablas HTML

Esto merece alguna explicación: (1) la diferencia entre el número promedio de filas de las tablas HTML y el número promedio de filas de tablas de hojas de cálculo; (2) en nuestro proceso de clasificación, una fila dada se clasifica como «metadatos no relacionales» cuando la fila no puede ser identificada como de datos o de encabezado; (3) las tablas de hojas de cálculo tienen una mejor definición en términos de características, por ejemplo, las tablas dependen de propiedades encapsuladas dentro de archivos CSS.

3.4. Algoritmo basado en reglas para detección de tablas

El algoritmo basado en reglas fue aplicado a hojas en un conjunto muestra que contenía tablas con diferentes diseños y gráficos integrados. La Tabla 9 resume los resultados obtenidos, que analizaron un total de 1000 documentos de hoja de cálculo, detectaron 1481 tablas y clasificaron incorrectamente 141.

Documentos de hojas de cálculo	1000
Tablas	1481
Tablas clasificadas incorrectamente	141
Tabla simple	700
Tabla múltiple	158

Tabla 9. Resultados para el algoritmo de detección basado en reglas

El algoritmo falló para tablas múltiples con separadores internos que son menores que el umbral definido. En ese caso, el algoritmo consideraría las dos tablas como una tabla simple. Asimismo, no reconocería correctamente tablas cuando las celdas no tengan atributos o separadores (por ejemplo, una tabla sin bordes, sin formato de letra, sin colores de fondo y sin filas vacías que separen encabezados y el título de la tabla) y no descubrió tablas donde el número de celdas vacías a la derecha e izquierda es extremadamente grande.

3.4.1. Experimentos con el algoritmo de detección de tabla remover y conquistar

El algoritmo Remover y conquistar (RAC) fue aplicado al mismo conjunto de datos. Este algoritmo detectó tablas que no pudieron ser reconocidas por el algoritmo basado en reglas y maximizó la coincidencia entre la tabla propuesta P y la tabla verdadera T, lo cual es equivalente a maximizar el número de celdas que ellas tienen en común y minimizar el número de celdas por las cuales difieren. La Tabla 10 muestra los resultados al comparar con el algoritmo 1, donde se observa que el número de tablas clasificadas incorrectamente se redujo y el número de tablas múltiples detectadas aumentó.

Documentos de hojas de cálculo	1000
Tablas	1481
Tablas clasificadas incorrectamente	141
Tabla simple	650
Tabla múltiple	230

Tabla 10. Tablas reconocidas mediante RAC

3.5. Resultados de los algoritmos y descripción del ambiente

Antes de entrar en detalle acerca de los tiempos de ejecución de los algoritmos, se explicarán las características principales del ambiente: computadora portátil (PC) modelo Lenovo 80YH con 8 GB de memoria RAM, procesador Intel(R) Core i7-7500 con 2.70 GHz, tarjeta gráfica Intel(R) 620 con 128 MB de memoria, sistema operativo Windows 10 Home de 64 bits, la Tabla 11 muestra los tiempos de ejecución de cada uno de los algoritmos.

Algoritmo	Tiempo de ejecución (s)	CPU (%)	Memoria(%)
Remover y conquistar	114,28	4,3	1,3
Basado en reglas	69,19	3,7	1
Campo aleatorio condicional	376,57	11,5	2,5

Table 11. Execution Time to the Tables Recognition Algorithms

4. Conclusiones

En este trabajo se ha descrito la implementación de tres algoritmos para clasificar filas de una tabla y reconocer tablas en documentos de hojas de cálculo, respectivamente. Se realizaron experimentos para probar el desempeño del clasificador de filas de tablas utilizando tablas HTML y de hojas de cálculo. Los experimentos muestran que el clasificador obtuvo excelentes resultados para ambos tipos de tablas. Asimismo, se aplicó una validación cruzada k-fold donde se obtuvieron resultados similares a los otros experimentos reportados en [20].

En resumen, las contribuciones de este trabajo fueron:

- Un clasificador de filas de tabla, aplicable tanto a tablas HTML como a tablas de hojas de cálculo.
- Experimentos para validar el clasificador.
- Dos conjuntos de datos que contienen tablas HTML y de hojas de cálculo anotadas, para entrenar y validar clasificadores de filas de tablas.
- La implementación de dos algoritmos para el reconocimiento de tablas en documentos de hojas de cálculo.

Como trabajo futuro, se propone incrementar el número de instancias y clases en nuestros conjuntos de datos y agregar más características complejas. Se espera que los CRF también puedan ser aplicados a otras tareas de clasificación no tabular, que involucren contenido con varios formatos y diseños. En general, los CRF pueden ayudar en la construcción de sistemas de extracción de información genérica.

References

[1] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri, "Infogather: Entity augmentation and attribute discovery by holistic matching with web tables," in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 97–108. [Online]. Available: https://doi.org/10.1145/2213836.2213848

[2] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: Exploring the power of tables on the web," Proc. VLDB Endow., vol. 1, no. 1, pp. 538–549, Aug. 2008. [Online]. Available: https://doi.org/10.14778/1453856.1453916

[3] E. Koci, M. Thiele, O. Romero, and W. Lehner, "Table identification and reconstruction in spreadsheets," in Advanced Information Systems Engineering, E. Dubois and K. Pohl, Eds. Cham: Springer International Publishing, 2017, pp. 527–541.

[4] P. Venetis, A. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu, "Recovering semantics of tables on the web," Proc. VLDB Endow., vol. 4, no. 9, pp. 528–538, Jun. 2011. [Online]. Available: https://doi.org/10.14778/2002938.2002939

[5] G. Limaye, S. Sarawagi, and S. Chakrabarti, "Annotating and searching web tables using entities, types and relationships," Proc. VLDB Endow., vol. 3, no. 1–2, pp. 1338–1347, Sep. 2010. [Online]. Available: https://doi.org/10.14778/1920841.1921005

[6] T. F. Varish Mulwad and A. Joshi, "Generating Linked Data by Inferring the Semantics of Tables," in Proceedings of the First International Workshop on Searching and Integrating New Web Data Sources, September 2011, co-located with VLDB 2011. [Online]. Available: https://bit.ly/3p8s1q0

[7] A. S. Corrêa and P.-O. Zander, "Unleashing tabular content to open data: A survey on pdf table extraction methods and tools," in Proceedings of the 18th Annual International Conference on Digital Government Research, ser. dg.o '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 54–63. [Online]. Available: https://doi.org/10.1145/3085228.3085278

[8] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files." [Online]. Available: https://bit.ly/3k2ejBa

[9] Y. Liu, P. Mitra, and C. L. Giles, "Identifying table boundaries in digital documents via sparse line detection," in CIKM '08, 2008. [Online]. Available: https://bit.ly/369nWcm

[10] T. Kieninger, "Table structure recognition based on robust block segmentation," 1998, pp. 22–32. [Online]. Available: https://bit.ly/38k4YT9

[11] M. Zhang and K. Chakrabarti, "Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables," in Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 145–156. [Online]. Available: https://doi.org/10.1145/2463676.2465276

[12] Z. Zhang, "Towards efficient and effective semantic table interpretation," in The Semantic Web – ISWC 2014, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandecic, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds. Cham: Springer International Publishing, 2014, pp. 487–502. [Online]. Available: https://doi.org/10.1007/978-3-319-11964-9_31

[13] H. Masuda and S. Tsukamoto, "Recognition of html table structure," 2004. [Online]. Available: https://bit.ly/3p8xL2Q

[14] J. Fang, P. Mitra, Z. Tang, and C. L. Giles, "Table header detection and classification," in AAAI, 2012. [Online]. Available: https://bit.ly/2IcT3vy

[15] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, ser. SIGIR '03. New York, NY, USA: Association for Computing Machinery, 2003, pp. 235–242. [Online]. Available: https://doi.org/10.1145/860435.860479

[16] I. A. Doush and E. Pontelli, "Detecting and recognizing tables in spreadsheets," in Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, ser. DAS '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 471–478. [Online]. Available: https://doi.org/10.1145/1815330.1815391

[17] E. Koci, M. Thiele, W. Lehner, and O. Romero, "Table recognition in spreadsheets via a graph representation," in 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 2018, pp. 139–144. [Online]. Available: https://doi.org/10.1109/DAS.2018.48

[18] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in Proceedings of the Eighteenth International Conference on Machine Learning, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: https://bit.ly/3lbW1yE

[19] J. L. Solé, Book review: Pattern recognition and machine learning. Cristopher M. Bishop. Information Science and Statistics. Springer, 2007. [Online]. Available: https://bit.ly/3l7doRq

[20] M. D. Adelfio and H. Samet, "Schema extraction for tabular data on the web," Proc. VLDB Endow., vol. 6, no. 6, pp. 421–432, Apr. 2013. [Online]. Available: https://doi.org/10.14778/2536336.2536343