User guide

Getting Started with ContentMine table processing software

ContentMine norma 0.7.0-alpha (TableClipper)

The ContentMine document processing system is known as **norma**. norma provides a collection of transforms from documents (full or partial) to structured and usable data.

This document introduces the commands used to transform tables to semantic format.

System requirements

- Java 1.8 runtime (Oracle or OpenJDK)
- OS: Linux, MacOS X, Windows 8+

The following instructions assume a command-line shell such as bash Linux or MacOS X or git-bash, cygwin or Windows Powershell.

Version

This document is intended for use with the current snapshot release `norma-0.7.0-alpha`

Running norma on an existing corpus

To get the binary either, use git:

```
  git clone https://github.com/ContentMine/releases/norma/norma-0.7.0-alpha
```

or download the following and and unzip it:

https://github.com/ContentMine/releases/norma/norma-0.7.0-alpha

This release package has the following contents:

- Binaries:
    - norma-0.7.0-alpha 2012 – single jar file including dependencies

and

    - norma executables (in bin) with jar dependencies (in repo)
- Example CSS referencing HTML5 attributes and structural markup
- Example document from the Cochrane online corpus (Open Access)

norma-0.7.0-alpha/bin/norma is the latest norma binary (version norma-0.7.0-alpha)

Usage

Transforming PDFs to HTML and CSV

The norma software is a toolkit supporting transformations between different document formats. These transformations also extract and represent semantic structure and normalise numeric and symbolic values.

The workflow to extract tables from PDF documents to HTML and CSV consists of four norma transforms: **pdf2svg, cropbox, svgtable2html** and **svgtable2csv**.

The bash script `clipandtransform.sh` gives an example of using norma workflow on the command line to convert all the tables from one PDF document.

The steps in the workflow are as a follows:

- **makeProject** – create the results directory structure

Each PDF document contains multiple pages, including text, images, tables. diagrams etc. This means that processing each individual PDF file results in multiple output files. To organise these result files, for each document, norma creates a directory tree with a consistent structure, called a **CTree**.

For convenience it is useful to group PDF documents into a corpus. norma groups the CTrees generated for one or more documents in a corpus into a **CProject**.

Path regexes Norma uses regexes on the command line to list files within the CTree and CProject.

Note that these regexes should always use UNIX/XPath style, with forward-slashes as a path separator. Conversion to the current platform is handled automatically. Also note that forward-slashes should not be escaped in these strings as this will cause the filter to fail on Windows.

The following shows a workflow from PDF document and crowd-sourced coordinates in mm to HTML5 and CSV outputs.

```
${NORMA} --project TestCorpus --fileFilter ".*/(.*)\\.pdf"
--makeProject "(\\1)/fulltext.pdf"
```

- **pdf2svg** – converts PDF document(s) into individual pages in SVG format

```
${NORMA} --project TestCorpus --input fulltext.pdf --outputDir
TestCorpus --transform pdf2svg
```

- **cropbox** – use coordinates and page numbers to clip the tables from the SVG pages, omitting other content.

For example, given the following data from EPPI:

PDF:

http://eppi.ioe.ac.uk/pdfs4crowd/16515705.pdf

and coordinates data for its two tables:

```
{
   "H1516291135139":{
      "widthmm":178.5,
      "widthpx":674.4,
      "heightmm":97.5,
      "heightpx":368.8,
      "page":5,
      "topmm":26,
      "toppx":99.2,
      "originalscale":1.25,
      "leftmm":17.5,
      "leftpx":65.6,
      "tabletype":"Participant Characteristics"
   },
   "H1516291144796":{
      "widthmm":180.5,
      "widthpx":683.2,
      "heightmm":41,
      "heightpx":156,
      "page":5,
      "topmm":218.5,
      "toppx":825.6,
      "originalscale":1.25,
      "leftmm":16,
      "leftpx":60,
      "tabletype":"Results"
   }
}
```

Using the mm values, the corresponding norma commandline to call the cropbox transform for the first table would be as follows:

```
${NORMA} --project TestCorpus --cropbox x0 17.5 y0 26 width 178.5
height 97.5 ydown units mm --pageNumbers 5 --output
tables/table1516291135139/table.svg
```

Note that the table number is not extracted automatically from the text within the document.   Table numbers are long (64-bit) integers and may contain up to 19 decimal digits. In this example the numerical part of the Cochrane Crowd ID (H1516291135139) for the table has been used.

The cropbox parameters are used as follows:

  - **x0** – x coordinate of top-left corner (in mm)
  - **y0** – y coordinate of top-left corner (in mm)
  - **width** – width of table (in mm)
  - **height** – height of table (in mm)
  - **ydown** – direction of y coordinate system

    ydown means that y=0 at the top of the page and coordinates increase towards the page bottom. This is the more common coordinate system for PDF processing and should be used as the default. Use **yup** instead if y=0 at the bottom of the page and coordinate increase upwards.

  - **units mm** – the units of the x0, y0, width and height parameters. The recommended default is to use mm.

- **svgtable2html** – transform clipped tables in SVG format into semantically structured XML-compliant HTML5

All tables in paper are now in the CTree as SVG. The svgtable2html transform extracts and normalises individual numerical values and determines semantic table structure and represents this in HTML structures and attributes

```
${NORMA} --project TestCorpus --fileFilter
"^.*tables/table(\\d+)/table(_\\d+)?\\.svg" --outputDir TestCorpus
--transform svgtable2html
```

This will generate HTML5 files stored under the TestCorpus `directory tree.`

The resulting HTML files will appear as files named `table.svg.html` in the CTree directory in the **table<*Number*>** directory for each table.

To ensure separation of data and presentation, the HTML output is not visually styled. The semantic structure of the table is represented using standard HTML table components (thead, tbody, tfoot), classes (e.g., subtable) and HTML5 data-attributes. All of these can be styled using CSS or utilised for further processing.

- **svgtable2csv** – transform clipped tables in SVG format into CSV format

To produce output in CSV format, a similar command line is used:

```
${NORMA} --project TestCorpus --fileFilter
"^.*tables/table(\\d+)/table(_\\d+)?\\.svg" --outputDir TestCorpus
```

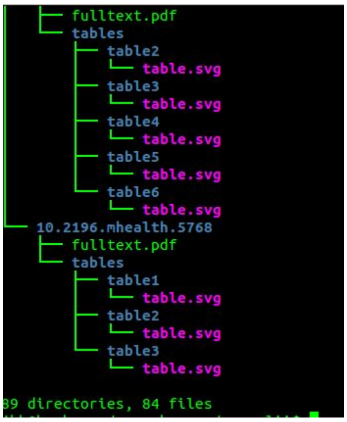
```
--transform svgtable2csv
```

The resulting CSV files will appear as files named `table.svg.csv` in the CTree directory in the table*Number* directory for each table.

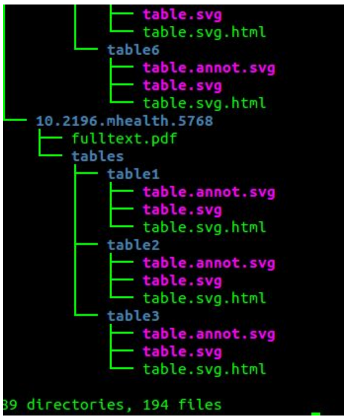
Expected behaviour – details

The name of each document is output to the command line as norma runs. More detailed output from the run including any errors are logged to `norma.log` in the current directory. Each day's log is rolled over into a dated log file so that the current day's log does not grow too large.

A successful run will output resulting HTML files (`table`*`Number`*`/table.svg.html`) and some intermediate files capturing the system's analysis of the table .



```
│   ├── fulltext.pdf
│   └── tables
│       ├── table2
│       │   └── table.svg
│       ├── table3
│       │   └── table.svg
│       ├── table4
│       │   └── table.svg
│       ├── table5
│       │   └── table.svg
│       └── table6
│           └── table.svg
└── 10.2196.mhealth.5768
    ├── fulltext.pdf
    └── tables
        ├── table1
        │   └── table.svg
        ├── table2
        │   └── table.svg
        └── table3
            └── table.svg

39 directories, 84 files
```



```
│       │   ├── table.svg
│       │   └── table.svg.html
│       └── table6
│           ├── table.annot.svg
│           ├── table.svg
│           └── table.svg.html
└── 10.2196.mhealth.5768
    ├── fulltext.pdf
    └── tables
        ├── table1
        │   ├── table.annot.svg
        │   ├── table.svg
        │   └── table.svg.html
        ├── table2
        │   ├── table.annot.svg
        │   ├── table.svg
        │   └── table.svg.html
        └── table3
            ├── table.annot.svg
            ├── table.svg
            └── table.svg.html

39 directories, 194 files
```

Demo

The release package includes a demo using an Open Access paper from the Cochrane crowd corpus and the coordinates an measurements supplied with it.

Demo contents:

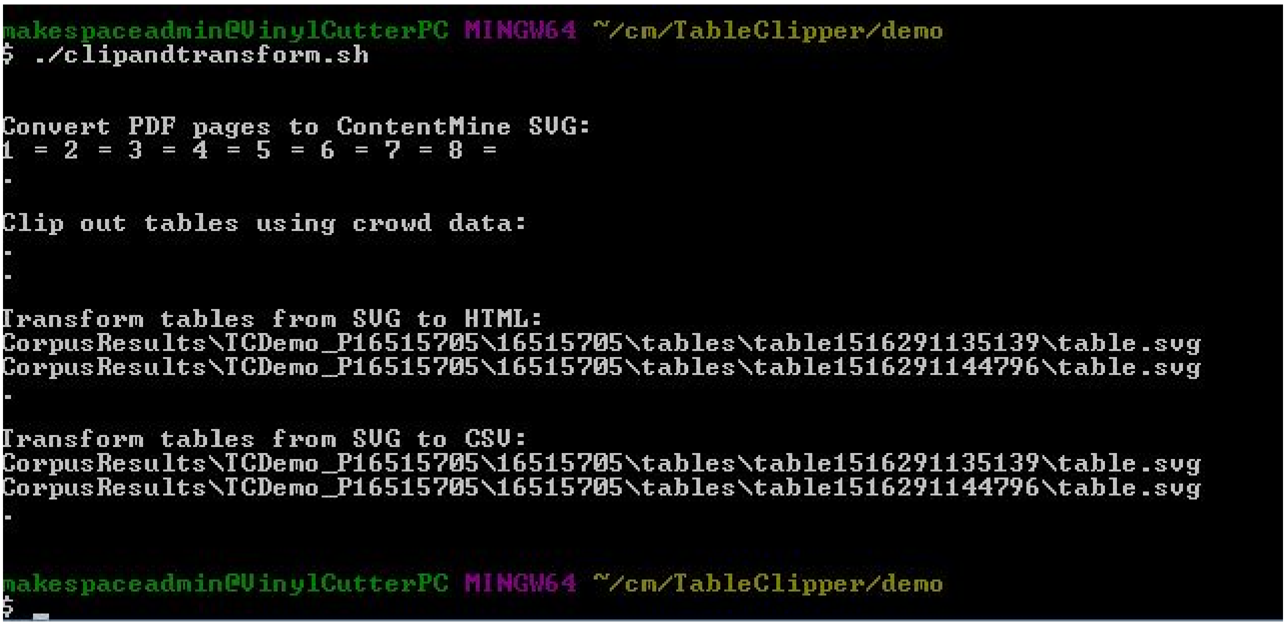
- `clipandtransform.sh`
- `cochrane_corpus/16515705.pdf`
- `cmtblstructure.css` (example reference stylesheet for HTML structures)

Running the demo

The demo script `clipandtransform.sh` can be run on Linux under bash, or Windows under git-bash or Powershell. The screenshots show usage under Windows 8.1 x64 using git-bash.

In the demo directory, run `clipandtransform.sh`.

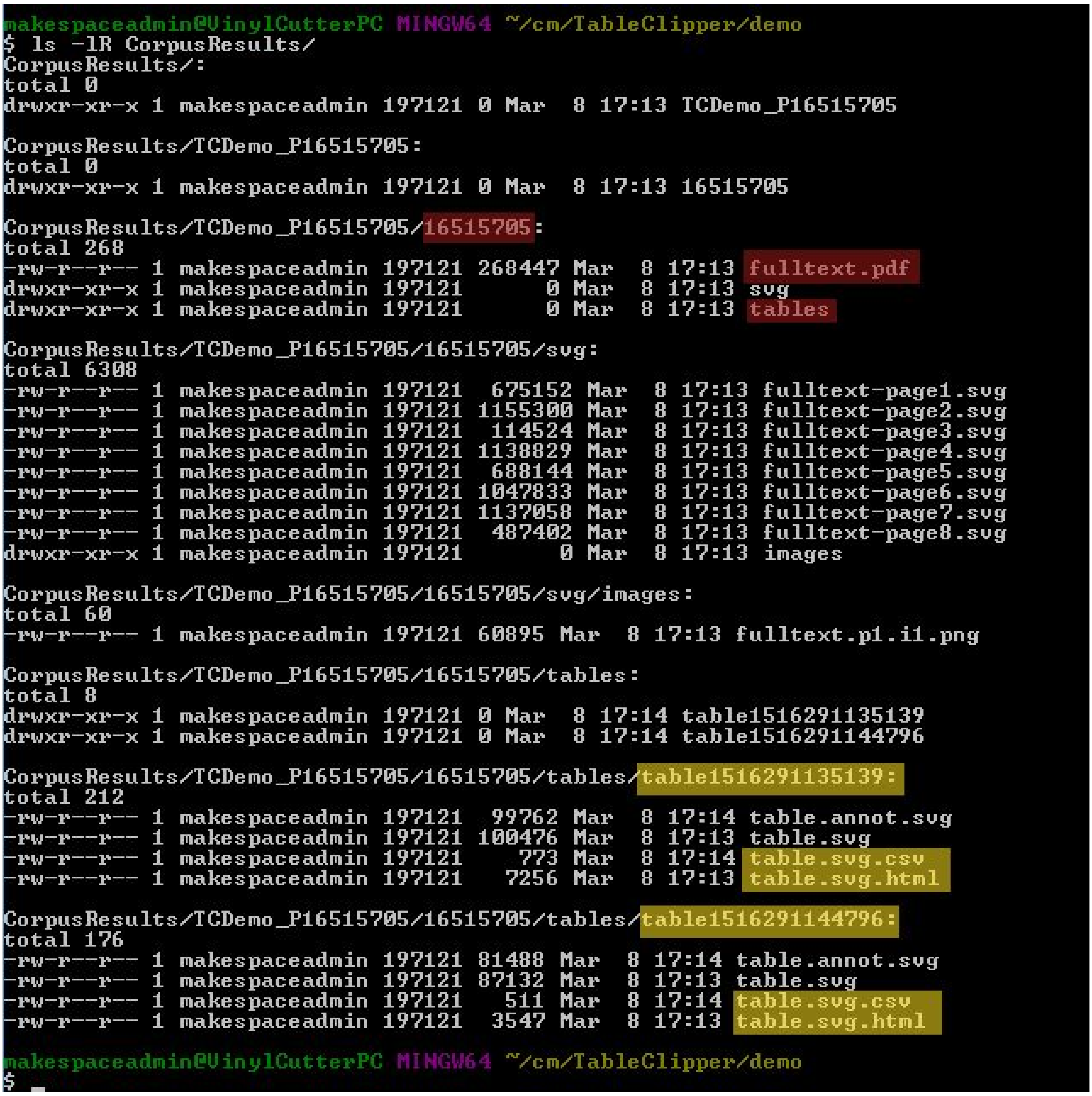
A successful run will produce the following confirmation output, including output from the `norma`:



```
makespaceadmin@VinylCutterPC MINGW64 ~/cm/TableClipper/demo
$ ./clipandtransform.sh


Convert PDF pages to ContentMine SVG:
1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 =
-

Clip out tables using crowd data:
-
-

Transform tables from SVG to HTML:
CorpusResults\TCDemo_P16515705\16515705\tables\table1516291135139\table.svg
CorpusResults\TCDemo_P16515705\16515705\tables\table1516291144796\table.svg
-

Transform tables from SVG to CSV:
CorpusResults\TCDemo_P16515705\16515705\tables\table1516291135139\table.svg
CorpusResults\TCDemo_P16515705\16515705\tables\table1516291144796\table.svg
-


makespaceadmin@VinylCutterPC MINGW64 ~/cm/TableClipper/demo
$
```

In this demo we have used the project name `CorpusResults`. So this will contain all output files in HTML and CSV format, as well as intermediate files produced by the extraction and conversion phases, including SVG and PNG images.

After a successful run, the results of converting paper **`16515705.pdf`** to extract tables **H1516291135139** and **H1516291144796** are found within the **CTree** structure as follows. Highlighted in red are directories and files for paper 16515705 as a whole. The results of extracting and processing individual tables are stored under the **`tables`** subdirectory. Highlighted in yellow are the locations of the output files for each of the two tables.



```
makespaceadmin@VinylCutterPC MINGW64 ~/cm/TableClipper/demo
$ ls -lR CorpusResults/
CorpusResults/:
total 0
drwxr-xr-x 1 makespaceadmin 197121 0 Mar  8 17:13 TCDemo_P16515705

CorpusResults/TCDemo_P16515705:
total 0
drwxr-xr-x 1 makespaceadmin 197121 0 Mar  8 17:13 16515705

CorpusResults/TCDemo_P16515705/16515705:
total 268
-rw-r--r-- 1 makespaceadmin 197121 268447 Mar  8 17:13 fulltext.pdf
drwxr-xr-x 1 makespaceadmin 197121      0 Mar  8 17:13 svg
drwxr-xr-x 1 makespaceadmin 197121      0 Mar  8 17:13 tables

CorpusResults/TCDemo_P16515705/16515705/svg:
total 6308
-rw-r--r-- 1 makespaceadmin 197121  675152 Mar  8 17:13 fulltext-page1.svg
-rw-r--r-- 1 makespaceadmin 197121 1155300 Mar  8 17:13 fulltext-page2.svg
-rw-r--r-- 1 makespaceadmin 197121  114524 Mar  8 17:13 fulltext-page3.svg
-rw-r--r-- 1 makespaceadmin 197121 1138829 Mar  8 17:13 fulltext-page4.svg
-rw-r--r-- 1 makespaceadmin 197121  688144 Mar  8 17:13 fulltext-page5.svg
-rw-r--r-- 1 makespaceadmin 197121 1047833 Mar  8 17:13 fulltext-page6.svg
-rw-r--r-- 1 makespaceadmin 197121 1137058 Mar  8 17:13 fulltext-page7.svg
-rw-r--r-- 1 makespaceadmin 197121  487402 Mar  8 17:13 fulltext-page8.svg
drwxr-xr-x 1 makespaceadmin 197121       0 Mar  8 17:13 images

CorpusResults/TCDemo_P16515705/16515705/svg/images:
total 60
-rw-r--r-- 1 makespaceadmin 197121 60895 Mar  8 17:13 fulltext.p1.i1.png

CorpusResults/TCDemo_P16515705/16515705/tables:
total 8
drwxr-xr-x 1 makespaceadmin 197121 0 Mar  8 17:14 table1516291135139
drwxr-xr-x 1 makespaceadmin 197121 0 Mar  8 17:14 table1516291144796

CorpusResults/TCDemo_P16515705/16515705/tables/table1516291135139:
total 212
-rw-r--r-- 1 makespaceadmin 197121  99762 Mar  8 17:14 table.annot.svg
-rw-r--r-- 1 makespaceadmin 197121 100476 Mar  8 17:13 table.svg
-rw-r--r-- 1 makespaceadmin 197121    773 Mar  8 17:14 table.svg.csv
-rw-r--r-- 1 makespaceadmin 197121   7256 Mar  8 17:13 table.svg.html

CorpusResults/TCDemo_P16515705/16515705/tables/table1516291144796:
total 176
-rw-r--r-- 1 makespaceadmin 197121 81488 Mar  8 17:14 table.annot.svg
-rw-r--r-- 1 makespaceadmin 197121 87132 Mar  8 17:13 table.svg
-rw-r--r-- 1 makespaceadmin 197121   511 Mar  8 17:14 table.svg.csv
-rw-r--r-- 1 makespaceadmin 197121  3547 Mar  8 17:13 table.svg.html

makespaceadmin@VinylCutterPC MINGW64 ~/cm/TableClipper/demo
$ 
```

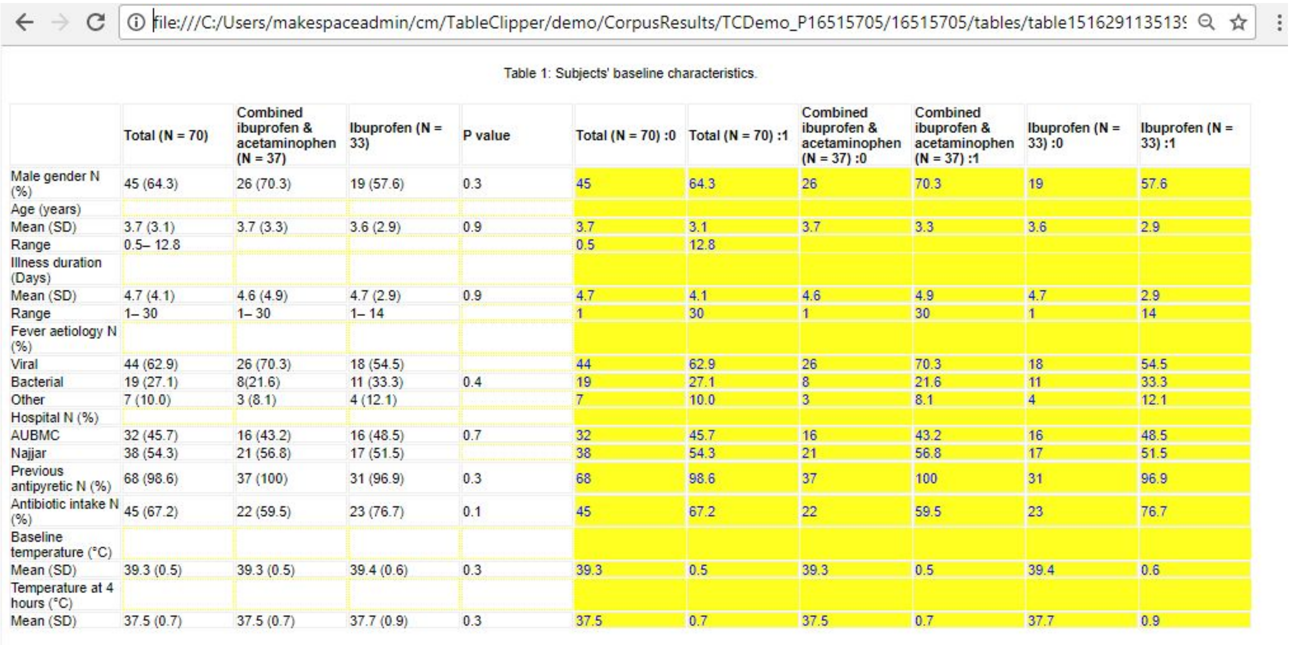
HTML file format for semantically structured data

The HTML makes use of standard table-structuring elements such as caption, thead, tfoot and tbody. Data attributes and classes are also used to capture the results of analysing the table structure and marking up extracted results.

Although the table structure is kept separate from styling, the demo contains a CSS3 file which can be used to style based on the HTML.

Example:

Table H1516291135139:



file:///C:/Users/makespaceadmin/cm/TableClipper/demo/CorpusResults/TCDemo_P16515705/16515705/tables/table15162911351З

Table 1: Subjects' baseline characteristics.

	Total (N = 70)	Combined ibuprofen & acetaminophen (N = 37)	Ibuprofen (N = 33)	P value	Total (N = 70) :0	Total (N = 70) :1	Combined ibuprofen & acetaminophen (N = 37) :0	Combined ibuprofen & acetaminophen (N = 37) :1	Ibuprofen (N = 33) :0	Ibuprofen (N = 33) :1
Male gender N (%)	45 (64.3)	26 (70.3)	19 (57.6)	0.3	45	64.3	26	70.3	19	57.6
Age (years)										
Mean (SD)	3.7 (3.1)	3.7 (3.3)	3.6 (2.9)	0.9	3.7	3.1	3.7	3.3	3.6	2.9
Range	0.5– 12.8				0.5	12.8				
Illness duration (Days)										
Mean (SD)	4.7 (4.1)	4.6 (4.9)	4.7 (2.9)	0.9	4.7	4.1	4.6	4.9	4.7	2.9
Range	1– 30	1– 30	1– 14		1	30	1	30	1	14
Fever aetiology N (%)										
Viral	44 (62.9)	26 (70.3)	18 (54.5)		44	62.9	26	70.3	18	54.5
Bacterial	19 (27.1)	8(21.6)	11 (33.3)	0.4	19	27.1	8	21.6	11	33.3
Other	7 (10.0)	3 (8.1)	4 (12.1)		7	10.0	3	8.1	4	12.1
Hospital N (%)										
AUBMC	32 (45.7)	16 (43.2)	16 (48.5)	0.7	32	45.7	16	43.2	16	48.5
Najjar	38 (54.3)	21 (56.8)	17 (51.5)		38	54.3	21	56.8	17	51.5
Previous antipyretic N (%)	68 (98.6)	37 (100)	31 (96.9)	0.3	68	98.6	37	100	31	96.9
Antibiotic intake N (%)	45 (67.2)	22 (59.5)	23 (76.7)	0.1	45	67.2	22	59.5	23	76.7
Baseline temperature (°C)										
Mean (SD)	39.3 (0.5)	39.3 (0.5)	39.4 (0.6)	0.3	39.3	0.5	39.3	0.5	39.4	0.6
Temperature at 4 hours (°C)										
Mean (SD)	37.5 (0.7)	37.5 (0.7)	37.7 (0.9)	0.3	37.5	0.7	37.5	0.7	37.7	0.9

In this example columns containing multiple numerical values (such as the first column headed 'Total (N=70)') are split and the individual values added to each row as supplemental columns. Due to the lack of consistent column-naming across publications, the headings of the split columns are of the form <**Original Column Name>: <value index>** where value indexes start at 0.
These additional columns of extracted values are also output in the CSV.

norma uses HTML5 data-attributes to mark up different kinds of content.

This extract from HTML5 source shows the header and first row of the table. This shows how supplemental column headers ('supp-header') and column value cells holding observation values ('supp-obs') are marked up in the output HTML and can thus be styled or processed further.

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="UTF-8" />
  </head>
  <body>
    <table class="table">
      <caption>Table 1: Subjects' baseline characteristics.</caption>
      <thead>
        <tr data-tblrole="columnheaderrow">
          <th />
          <th class="cell" data-cellminx="210.000" data-cellmaxx="265.000">Total (N = 70) </th>
          <th class="cell" data-cellminx="293.000" data-cellmaxx="395.000">Combined ibuprofen &amp; acetaminophen (N = 37) </th>
          <th class="cell" data-cellminx="424.000" data-cellmaxx="495.000">Ibuprofen (N = 33) </th>
          <th class="cell" data-cellminx="516.000" data-cellmaxx="544.000">P value </th>
          <th data-role="supp-header">Total (N = 70) :0</th>
          <th data-role="supp-header">Total (N = 70) :1</th>
          <th data-role="supp-header">Combined ibuprofen &amp; acetaminophen (N = 37) :0</th>
          <th data-role="supp-header">Combined ibuprofen &amp; acetaminophen (N = 37) :1</th>
          <th data-role="supp-header">Ibuprofen (N = 33) :0</th>
          <th data-role="supp-header">Ibuprofen (N = 33) :1</th>
        </tr>
      </thead>
      <tbody>
        <tr data-rowminx="91.400" data-rowminy="133.800">
          <td class="cell" data-cellminx="91.400" data-cellminy="133.800">Male gender N
(%)</td>
          <td class="cell" data-cellminx="223.340" data-cellminy="133.800">45 (64.3)</td>
            <td class="cell" data-cellminx="329.113" data-cellminy="133.800">26
(70.3)</td>
            <td class="cell" data-cellminx="445.196" data-cellminy="133.800">19
(57.6)</td>
            <td class="cell" data-cellminx="525.694" data-cellminy="133.800">0.3</td>
            <td data-role="supp-obs" class="cell">45</td>
            <td data-role="supp-obs" class="cell">64.3</td>
            <td data-role="supp-obs" class="cell">26</td>
            <td data-role="supp-obs" class="cell">70.3</td>
            <td data-role="supp-obs" class="cell">19</td>
            <td data-role="supp-obs" class="cell">57.6</td>
          </tr>

...
       </tbody>
       <tfoot>
            ...
       </tfoot>
    </table>
  </body>
</html>
```