

Invoice Extraction Checker Documentation

Version 1.0.0

15th June 2019

Table of Contents

1	Introduction.....	3
1.1	Templates.....	3
1.2	Installing the application.....	3
1.3	Running the application.....	4
2	Writing a template.....	5
2.1	Naming the template.....	6
2.2	Keywords – Applying a template.....	6
2.3	Fields – Picking values for a given name.....	7
2.3.1	Name.....	7
2.3.2	Location.....	8
2.3.2.1	Right.....	8
2.3.2.2	Second-right.....	9
2.3.2.3	Bottom.....	9
2.3.2.4	Regex.....	10
2.3.2.5	Regex right.....	10
2.3.3	Identifier.....	11
2.3.4	Ordinal.....	12
2.4	Picking Line Items.....	13
2.4.1	Table Line Items.....	13
2.4.1.1	Both Horizontal and Vertical Lines.....	14
2.4.1.2	Vertical Lines but no Horizontal Lines.....	16
2.4.1.3	Horizontal Lines but no Vertical Lines.....	18
2.4.1.4	No Vertical Lines or Horizontal Lines.....	20
2.4.2	Regex Line Items.....	22
2.5	Checking extracted values.....	23
3	Customization using plugins.....	26

1 Introduction

This project is based on invoice2data which extracts data from invoice pdf files using user written regular expressions. It is observed that quite some invoices cannot be extracted just using the regular expression approach since pdf text is organized not as lines of text rather as boxes with co-ordinates. Field values are placed at locations like to the right, bottom of a field so on. Line items can also placed in tables. This solution tries to use other techniques along with regular expressions to solve the problem.

The solution is written in python and provides the following capabilities

- 1 Extract Field values (Based on regex or location based - right, bottom ...)
- 2 Table based line item extraction (regex, vertical or horizontal table lines ...)
- 3 Check if the extracted line item Total values matches the Sum Total.

1.1 Templates

Templates which are in json format are used to tell the application how to pick field values (regex, top, bottom of field name) and also how line items are present in the invoice (With or Without Horizontal Vertical Lines, Line item columns...).

1.2 Installing the application

- Install python 2.7 (Anaconda)
- pip/conda install the following libraries json, re, deepcopy, configparser, logging, pdfminer, argparse, sortedcontainers, sets

1.3 Running the application

From the src folder run,

```
python Main.py --dump <Dump file name> --template  
<Template Folder> --file <PDF file name> --output <Output JSON file>
```

```
$ls ..  
data LICENSE output README.md schema src template  
$python Main.py --dump dump.txt --template ../template/ --file "../data/FlipKart-ShreyasRetail.pdf"  
--output "../output/FlipKart-ShreyasRetail.json"  
2019-06-03 22:04:31,896 root INFO ***** invoice-extractor-checker STARTED :-) *****  
2019-06-03 22:04:31,897 root INFO Parsing file ../data/FlipKart-ShreyasRetail.pdf  
2019-06-03 22:04:33,363 root INFO Picked template file ../template/FlipKart-ShreyasRetail.json  
2019-06-03 22:04:33,368 root INFO Total fields : Configured = 3, Extracted = 3  
2019-06-03 22:04:33,374 root INFO Total Line Items Extracted = 16  
2019-06-03 22:04:33,375 root INFO Check Status Match = {'status': True}  
2019-06-03 22:04:33,377 root INFO ***** invoice-extractor-checker COMPLETED *****  
$
```

2 Writing a template

Template is written in json format and contains the name of the template, keywords that would be used to match the given template with a invoice pdf, field names and information on how to pick the corresponding field values, the columns in line items.

Before writing the template, it is a must that two files are held as reference. The first one being the invoice pdf itself, which would be used to identify the location of fields, columns of the line items so on. The second one that would come handy is a dump of the pdf that is created by this application. Text is represented as text boxes with x & y coordinates and the dump contains the location of the different text boxes in the given pdf.

Below you see the start of the first page dump, followed by a text box at location y 548.59 and location starting and ending at x0 and x1 respectively.

```
=====
PAGE # 0
=====
548.59
(x0= 96.64,x1= 199.67) |Tax Invoice/Bill of Supply|
```

Below you can see 4 text boxes aligned horizontal next to each other.

```
461.31
(x0= 9.38,x1= 44.77) |Invoice Number|
(x0= 44.78,x1= 93.24) |: FAB05L2000005510|
(x0= 151.01,x1= 178.74) |Invoice Date|
(x0= 178.76,x1= 206.34) |: 02-04-2019|
```

2.1 Naming the template

Name of this template must be unique across all the templates that are part of the system.

```
{
  "name": "Flipkart",
  "keywords": [
    "Shreyash Retail Private Limited"
  ],
  "fields": [
    {
```

2.2 Keywords - Applying a template

A list of all the text that would exist in the pdf. If all matches, apply this template i.e., use the fields cum line item definition to extract information from the pdf.

Template	<pre>{ "name": "Flipkart", "keywords": ["Shreyash Retail Private Limited"], "fields": [{</pre>
PDF	<div style="border: 1px solid black; padding: 10px;"> <div style="text-align: center; border-bottom: 1px solid black; margin-bottom: 10px;"> Tax Invoice/Bill of Supply </div> <p>Sold By: Shreyash Retail Private Limited ,</p> <p>Ship Address: No 3/1, 4/2, Kodipalya Dasanapura (Hobli), Bangalore North - 562162, Banç KA GSTIN - 29AAXCS0655F1ZU PAN - AAXCS0655F</p> </div>

2.3 Fields - Picking values for a given name

A list of all the fields that need to be extracted along with information about how to extract them.

Template	<pre>"fields":[{ "name":"Invoice Number", "location":"right", "identifier":"Invoice Number:" }, { "name":"Invoice Date", "location":"right", "identifier":"Invoice Date:" }, { "name":"Total", "location":"second-right", "identifier":"Summary" }],</pre>
----------	--

In the above example, there are 3 fields to be extracted. The name of field that would be extracted, the location of the field's value and information on how to identify the field in the pdf.

2.3.1 Name

Name of the field. The same name would be used in the output extracted json file.

2.3.2 Location

Provide the location detail for the particular field. The following values are possible

2.3.2.1 Right

Template	{ "name":"Invoice Number", "location":"right", }
PDF	Order: 0D1150982639/5935000 Invoice Number: FAB05L2000005510 Payment method: PREPAID Total Items: 15
Dump	461.31 (x0= 9.38,x1= 44.77) Invoice Number (x0= 44.78,x1= 93.24) : FAB05L2000005510 (x0= 151.01,x1= 178.74) Invoice Date (x0= 178.76,x1= 206.34) : 02-04-2019
Extraction	{ "fields": { "Invoice Number": ": FAB05L2000005510", ... }

Template	{ "name":"Invoice Number", "location":"right", }
PDF	Invoice Number: 10551/000003228
Dump	680.34 (x0= 413.98,x1= 480.73) Invoice Number: (x0= 502.47,x1= 567.72) 10551/000003228
Extraction	{ "fields": { "Invoice Number": "10551/000003228", , ... }

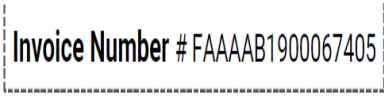
2.3.2.2 Second-right

Template	{ "name":"Tax", "location":"second-right", "identifier":"TOTAL:" },				
PDF	<table><tr><td>TOTAL:</td><td></td><td>\$313,538.08</td><td>\$31,353.81</td></tr></table>	TOTAL:		\$313,538.08	\$31,353.81
TOTAL:		\$313,538.08	\$31,353.81		
Dump	196.49 (x0= 293.46,x1= 310.75) TOTAL: (x0= 382.62,x1= 409.00) \$313,538.08 (x0= 434.40,x1= 458.14) \$31,353.81				
Extraction	{ "fields": { ... "Tax": "\$31,353.81", ... }				


2.3.2.3 Bottom

Template	{ "name":"Subtotal", "location":"bottom", "identifier":"Net" },						
PDF	<table><tr><th>Net</th><th>GST</th><th>Amount Due</th></tr><tr><td>2,429.22</td><td>242.92</td><td>2,672.14</td></tr></table>	Net	GST	Amount Due	2,429.22	242.92	2,672.14
Net	GST	Amount Due					
2,429.22	242.92	2,672.14					
Dump	<div>565.19</div> <div>(x0= 406.45,x1= 420.95) Net </div> <div>(x0= 466.20,x1= 484.68) GST </div> <div>(x0= 510.70,x1= 564.19) Amount Due </div> <div>543.10</div> <div>(x0= 394.17,x1= 429.18) 2,429.22 </div> <div>(x0= 460.17,x1= 487.69) 242.92 </div> <div>(x0= 532.92,x1= 567.93) 2,672.14 </div>						
Extraction	"Tax": "242.92", "Total": "2,672.14", Subtotal": "2,429.22"						

2.3.2.4 Regex

Template	{ "name":"Invoice Number", "location":"regex", }
PDF	
Dump	802.84 (x0= 415.31,x1= 563.32) Invoice Number # FAAAAB1900067405
Extraction	{ "fields": { "Invoice Number": "FAAAB1900067405", ... }

2.3.2.5 Regex right

Template	{ "name":"Total", "location":"regex-right", "identifier":"Total Current Charges - Due .*" }
PDF	
Dump	520.52 (x0= 48.74,x1= 197.73) Total Current Charges - Due 10/23 (x0= 247.56,x1= 282.59) 2,167.88
Extraction	"fields": { ... "Total": "2,167.88", ... }

The other possibilities could be regex bottom, regex second-right

2.3.3 Identifier

This text tells how to identify the field in the pdf. Matching text could be Text “as is” or a regular expression match.

“as is”

Template	<pre>{ "name":"Subtotal", "location":"bottom", "identifier":"Net" },</pre>						
PDF	<table><tr><th>Net</th><th>GST</th><th>Amount Due</th></tr><tr><td>2,429.22</td><td>242.92</td><td>2,672.14</td></tr></table>	Net	GST	Amount Due	2,429.22	242.92	2,672.14
Net	GST	Amount Due					
2,429.22	242.92	2,672.14					

“regular expression”

Template	{ "name":"Total", "location":"regex-right", "identifier":"Total Current Charges - Due .*" }		
PDF	<table border="1"> <tbody> <tr> <td>Total Current Charges - Due 10/23</td><td>2,167.88</td></tr> </tbody> </table>	Total Current Charges - Due 10/23	2,167.88
Total Current Charges - Due 10/23	2,167.88		

2.3.4 Ordinal

Optional. Default first (1). Sometimes the identifier given could occur at multiple locations in the pdf file. Ordinal is used to tell the application the field value needs to be picked from what ordinal.

Template	<pre>"fields":[{ "name":"Invoice Number", "location":"bottom", "identifier":"Invoice Number", "ordinal":1 }, </pre>
PDF	<div>First occurrence</div> <div>Invoice Number</div> <div>70128</div> <div>Second occurrence</div> <div>Invoice Number70128</div>

2.4 Picking Line Items

Line items can either be of tabular format or simple lines which can be extracted using regular expressions.

2.4.1 Table Line Items

To extract line items we need to specify the list of all the columns in the line item, the alignment of text under each column, what all columns constitutes a start of a line item and whether line items have vertical and horizontal guide lines.

2.4.1.1 Both Horizontal and Vertical Lines

Template

```
{
  "name": "RBI",
  "keywords": [
    "NATIONAL ELECTRONIC FUND TRANSFER"
  ],
  "fields": [
    {
      "name": "MONTH",
      "location": "regex",
      "identifier": "NATIONAL ELECTRONIC FUND TRANSFER \\(NEFT\\) - (.*)"
    }
  ],
  "table_lineitems": {
    "horizontal_lines" : true,
    "vertical_lines" : true,
    "columns": [
      {
        "name": "Sr. No",
        "alignment": "center",
        "row_start": true
      },
      {
        "name": "BANK NAME",
        "alignment": "left",
        "row_start": true
      },
      {
        "name": "NO. OF OUTWARD \nTRANSACTIONS",
        "alignment": "right",
        "row_start": true
      },
      {
        "name": "AMOUNT \n(Rs. Million)",
        "alignment": "right",
        "row_start": true
      },
      {
        "name": "NO. OF INWARD \nTRANSACTIONS",
        "alignment": "right",
        "row_start": true
      },
      {
        "name": "AMOUNT \n(Rs. Million)",
        "alignment": "right",
        "row_start": true
      }
    ],
    "line_end": "Total \\(No. of transactions in mn and Amount in bn\\)"
  }
}
```

PDF

Sr. No	BANK NAME	NO. OF OUTWARD TRANSACTIONS	AMOUNT (Rs. Million)	NO. OF INWARD TRANSACTIONS	AMOUNT (Rs. Million)
1	ABHYUDAYA CO-OP BANK LTD	90326	4743.4	205152	7615.1
2	ABU DHABI COMMERCIAL BANK	2417	386.1	1582	1020.9
3	AHMEDABAD MERCANTILE COOP BANK	17150	1686.0	32179	2368.1
4	AHMEDNAGAR MERCHANTS CO-OP BANK LTD	8364	819.9	5418	1991.2

Invoice Extraction Checker

Dump	<pre> ===== PAGE # 0 ===== 765.26 (x0= 170.04,x1= 446.04) NATIONAL ELECTRONIC FUND TRANSFER (NEFT) - APRIL 2019 754.40 (x0= 312.90,x1= 427.89) TOTAL OUTWARD DEBITS 754.34 (x0= 464.28,x1= 591.78) RECEIVED INWARD CREDITS 741.92 (x0= 294.78,x1= 377.78) NO. OF OUTWARD (x0= 395.22,x1= 449.72) AMOUNT (x0= 458.58,x1= 531.58) NO. OF INWARD (x0= 552.18,x1= 604.18) AMOUNT 736.16 (x0= 15.66,x1= 42.16) Sr. No (x0= 142.50,x1= 197.49) BANK NAME 730.34 (x0= 299.46,x1= 370.96) TRANSACTIONS (x0= 389.28,x1= 440.28) (Rs. Million) (x0= 458.22,x1= 529.72) TRANSACTIONS (x0= 546.24,x1= 597.24) (Rs. Million) 717.44 (x0= 26.76,x1= 31.76) 1 717.04 (x0= 51.06,x1= 172.85) ABHYUDAYA CO-OP BANK LTD (x0= 353.15,x1= 378.25) 90326 (x0= 505.80,x1= 535.92) 205152 716.86 (x0= 420.24,x1= 447.83) 4743.4 (x0= 576.48,x1= 604.07) 7615.1 704.48 (x0= 26.76,x1= 31.76) 2 704.08 (x0= 51.06,x1= 180.40) ABU DHABI COMMERCIAL BANK (x0= 358.22,x1= 378.29) 2417 (x0= 515.88,x1= 535.96) 1582 703.90 (x0= 425.28,x1= 447.85) 386.1 (x0= 576.48,x1= 604.07) 1020.9 691.52 (x0= 26.76,x1= 31.76) 3 </pre>
Extraction	<pre> "lineitems": [{ "AMOUNT \n(Rs. Million)": "7615.1", "NO. OF OUTWARD \nTRANSACTIONS": "90326", "Sr. No": "1", "BANK NAME": "ABHYUDAYA CO-OP BANK LTD", "AMOUNT \n(Rs. Million)": "4743.4", "NO. OF INWARD \nTRANSACTIONS": "205152" }, { "AMOUNT \n(Rs. Million)": "1020.9", "NO. OF OUTWARD \nTRANSACTIONS": "2417", "Sr. No": "2", "BANK NAME": "ABU DHABI COMMERCIAL BANK", "AMOUNT \n(Rs. Million)": "386.1", "NO. OF INWARD \nTRANSACTIONS": "1582" }, { "AMOUNT \n(Rs. Million)": "2368.1", "NO. OF OUTWARD \nTRANSACTIONS": "17150", "Sr. No": "3", "BANK NAME": "AHMEDABAD MERCANTILE COOP BANK", "AMOUNT \n(Rs. Million)": "1686.0", "NO. OF INWARD \nTRANSACTIONS": "32179" },] </pre>

2.4.1.2 Vertical Lines but no Horizontal Lines

Template

```
"table_lineitems":{
  "horizontal_lines" : false,
  "vertical_lines" : true,
  "columns":[
    {
      "name":"Item Code",
      "row_start":false,
      "alignment":"center"
    },
    {
      "name":"Item Description",
      "row_start": false,
      "alignment":"center"
    },
    {
      "name":"Ordered",
      "row_start":true,
      "alignment":"right"
    },
    {
      "name":"Shipped",
      "row_start":true,
      "alignment":"right"
    },
    {
      "name":"UOM",
      "row_start":false,
      "alignment":"right"
    },
    {
      "name":"Unit Price",
      "row_start":true,
      "alignment":"center"
    },
    {
      "name":"Line Total",
      "row_start":true,
      "alignment":"center"
    }
  ],
  "line_end":"Ex Tax"
}
```

PDF

Item Code	Item Description	Ordered	Shipped	UOM	Unit Price	Line Total
	<i>Install Only</i>					
	20 L355D Centrifugal Fan	36.0	36.0		1,516.00	54,576.00
	30 Motorised Vent	68.0	68.0		357.00	24,276.00
	60 Miscellaneous Works	15.0	15.0		88.00	1,320.00

Dump	<pre> 520.72 (x0= 46.70,x1= 88.44) Item Code (x0= 176.40,x1= 243.69) Item Description (x0= 311.00,x1= 384.55) Ordered Shipped (x0= 394.80,x1= 416.24) UOM (x0= 441.30,x1= 480.91) Unit Price (x0= 516.40,x1= 555.83) Line Total 485.39 (x0= 117.70,x1= 160.55) Install Only 475.19 (x0= 117.70,x1= 209.89) 20 L355D Centrifugal Fan (x0= 328.70,x1= 344.64) 36.0 (x0= 369.70,x1= 385.64) 36.0 (x0= 460.80,x1= 492.62) 1,516.00 (x0= 532.20,x1= 568.61) 54,576.00 462.49 (x0= 117.70,x1= 187.44) 30 Motorised Vent (x0= 328.70,x1= 344.64) 68.0 (x0= 369.70,x1= 385.64) 68.0 (x0= 467.60,x1= 492.62) 357.00 (x0= 532.20,x1= 568.61) 24,276.00 449.59 (x0= 117.70,x1= 207.82) 60 Miscellaneous Works (x0= 328.70,x1= 344.64) 15.0 (x0= 369.70,x1= 385.64) 15.0 (x0= 472.10,x1= 492.62) 88.00 (x0= 536.80,x1= 568.62) 1,320.00 </pre>
Extraction	<pre> "lineitems": [{ "Item Description": " Install Only" }, { "Shipped": "36.0", "Item Description": "20 L355D Centrifugal Fan", "Ordered": "36.0", "Line Total": "54,576.00", "Unit Price": "1,516.00" }, { "Shipped": "68.0", "Item Description": "30 Motorised Vent ", "Ordered": "68.0", "Line Total": "24,276.00", "Unit Price": "357.00" }, { "Shipped": "15.0", "Item Description": "60 Miscellaneous Works ", "Ordered": "15.0", "Line Total": "1,320.00", "Unit Price": "88.00" }] </pre>

2.4.1.3 Horizontal Lines but no Vertical Lines

Template

```
"fields": [
  {
    "name": "Invoice Number",
    "location": "bottom",
    "identifier": "Invoice Number",
    "ordinal": 1
  }
],
"table_lineitems": {
  "horizontal_lines" : true,
  "vertical_lines" : false,
  "columns": [
    {
      "name": "Description",
      "row_start": true,
      "alignment": "center"
    },
    {
      "name": "UnitPrice",
      "row_start": true,
      "alignment": "center"
    },
    {
      "name": "GST",
      "row_start": false,
      "alignment": "center"
    },
    {
      "name": "Amount",
      "row_start": true,
      "alignment": "center"
    }
  ],
  "line_end": "Subtotal"
}
```

PDF

Description	UnitPrice	GST	Amount
Survey	0.00		0.00
Progressinvoiceforworkscompleted	13,621.74	18%	13,621.74
		Subtotal	13,621.74

Dump	<pre> 539.59 (x0= 31.20,x1= 73.77) Description (x0= 392.90,x1= 427.19) UnitPrice (x0= 468.90,x1= 483.34) GST (x0= 514.90,x1= 545.10) Amount 516.79 (x0= 31.20,x1= 55.64) Survey (x0= 413.50,x1= 429.44) 0.00 (x0= 548.10,x1= 564.04) 0.00 472.29 (x0= 31.20,x1= 160.07) Progressinvoiceforworkscompleted (x0= 393.00,x1= 429.40) 13,621.74 (x0= 467.70,x1= 483.22) 18% (x0= 527.60,x1= 564.01) 13,621.74 416.99 (x0= 452.70,x1= 483.34) Subtotal (x0= 527.60,x1= 564.01) 13,621.74 </pre>
Extraction	<pre> "lineitems": [{ "UnitPrice": "0.00", "Description": "Survey" }, { "Amount": "13,621.74", "UnitPrice": "13,621.74", "Description": "Progressinvoiceforworkscompleted", "GST": "18%" }] </pre>

2.4.1.4 No Vertical Lines or Horizontal Lines

Template	<pre>], "fields": [{ "name": "AVERAGE DAILY USE", "location": "bottom", "identifier": "AVERAGE DAILY USE" }], "table_lineitems": { "horizontal_lines" : false, "vertical_lines" : false, "columns": [{ "name": "CHARGES AND CREDITS", "alignment": "left", "row_start": true }, { "name": "QUANTITY", "alignment": "center", "row_start": true }, { "name": "RATE(Rs)", "alignment": "center", "row_start": true }, { "name": "AMOUNT(Rs)", "alignment": "center", "row_start": true }], "line_end": "TOTAL" } }</pre>																
PDF	<table><tr><th>CHARGES AND CREDITS</th><th>QUANTITY</th><th>RATE(Rs)</th><th>AMOUNT(Rs)</th></tr><tr><td>Supply Charge</td><td>30 Day/s</td><td>6.33456</td><td>190.0368</td></tr><tr><td>Slab 1</td><td>200 kWh</td><td>1.12345</td><td>224.69</td></tr><tr><td>Slab 2</td><td>520 kWh</td><td>2.12345</td><td>1104.194</td></tr></table>	CHARGES AND CREDITS	QUANTITY	RATE(Rs)	AMOUNT(Rs)	Supply Charge	30 Day/s	6.33456	190.0368	Slab 1	200 kWh	1.12345	224.69	Slab 2	520 kWh	2.12345	1104.194
CHARGES AND CREDITS	QUANTITY	RATE(Rs)	AMOUNT(Rs)														
Supply Charge	30 Day/s	6.33456	190.0368														
Slab 1	200 kWh	1.12345	224.69														
Slab 2	520 kWh	2.12345	1104.194														

Dump	<pre> 683.23 (x0= 42.50,x1= 154.04) CHARGES AND CREDITS (x0= 252.60,x1= 305.64) QUANTITY (x0= 340.20,x1= 386.59) RATE(Rs) (x0= 425.10,x1= 488.38) AMOUNT(Rs) </pre>
Extraction	<pre> { "fields": { "AVERAGE DAILY USE": "40.032 kwh" }, "lineitems": [{ "AMOUNT(Rs)": "190.0368 ", "RATE(Rs)": "6.33456", "CHARGES AND CREDITS": "Supply Charge", "QUANTITY": "30 Day/s" }, { "AMOUNT(Rs)": "224.69 ", "RATE(Rs)": "1.12345", "CHARGES AND CREDITS": "Slab 1", "QUANTITY": "200 kwh" }, { "AMOUNT(Rs)": "1104.194 ", "RATE(Rs)": "2.12345", "CHARGES AND CREDITS": "Slab 2", "QUANTITY": "520 kwh" }] } </pre>

2.4.2 Regex Line Items

Template	<pre>"regex_lineitems":{ "line_start" : "Line\\s+Code\\s+Description\\s+Qty\\s+Total", "line_end" : "Total before GST", "lines":["^(?P<Line>\\d+)\\s+(?P<Code>\\d*)\\s+(? P<Description>.+)\\s+(?P<Qty>\\d+(,\\d+)*((\\.\\d+)?))\\s+(? P<Total>\\d+(,\\d+)*((\\.\\d+)?))\$"], "columns":["Line", "Code", "Description", "Qty", "Total"] }</pre>										
PDF	<table><tr><th>Line</th><th>Code</th><th>Description</th><th>Qty</th><th>Total</th></tr><tr><td>10</td><td>33333</td><td>SOMETHING ABCD;ABCD;FLOOR/WALL/HALL;FLEXI</td><td>2</td><td>95.98</td></tr></table>	Line	Code	Description	Qty	Total	10	33333	SOMETHING ABCD;ABCD;FLOOR/WALL/HALL;FLEXI	2	95.98
Line	Code	Description	Qty	Total							
10	33333	SOMETHING ABCD;ABCD;FLOOR/WALL/HALL;FLEXI	2	95.98							
Extraction	<pre>"lineitems": [{ "Line": "10", "Code": "33333", "Description": "SOMETHING ABCD;ABCD;FLOOR/WALL/HALL;FLEXI", "Total": "95.98", "Qty": "2" }, </pre>										

2.5 Checking extracted values

Checker is used to check if the Total value provided in the invoice and the sum of all the Amount columns in the line items match.

A field name which contains the extracted Total and the Line Item column name needs to be provided in the template. Regular expressions can also be provided to extract only the numeric value from the field/line item values. The result indicating the check status (matches, did not match for a particular reason...) shows up in the output json.

Template	<pre> { "check": { "field": { "name": "Total" }, "lineitem": { "name": "Total\nAmt(Rs)" } } } </pre>
----------	--

Sample showcasing picking a part of a field/line item value using regular expression for further extraction validation.

Template	<pre> { "check": { "field": { "name": "Total", "regex": "(?P<Total>\\d+(,\\d+)*((\\.\\d+)?) " }, "lineitem": { "name": "Total\n(incl GST)", "regex": "(?P<Total>(-)?\\d+(,\\d+)*((\\.\\d+)?) " } } } </pre>
----------	---

PDF content (Line Item column with values, Field Total value - Only few rows showcased)

PDF	<div>Total Amt(Rs)</div> <div></div> <div>349.00</div> <div>193.00</div> <div>282.99</div> <div>189.00</div>
Extracted (fields)	<pre>"fields": { "Invoice Number": ": FAB05L2000005510", "Invoice Date": ": 02-04-2019", "Total": "2309.99" }, "Total\nAmt(Rs)": "349.00", "HSN\n(Tax%)": "04059020 \n(12.0)", "Qty": "1", "Item": "Nandini Pure Cow Ghee 1 L Pouch", "MRP\n(Rs)": "450.00" }, { "S.\nNo": "2", "Savings\n(Rs)": "102.00", "Total\nAmt(Rs)": "193.00", . }</pre>
Extracted (Status)	<pre>"checkstatus": { "match_status": { "status": true } }</pre>

Match failure samples

Conversion Errors when the data in either line item columns or field value are not floating point numbers.

Extracted (Status)	<pre>"checkstatus": { "match_status": { "status": false, "description": "Conversion error" } }</pre>
-----------------------	--

When either wrong or missing extractions occur,

Extracted (Status)	<pre>"checkstatus": { "match_status": { "status": false, "description": "313538.08 != 321858.71" } }</pre>
-----------------------	--

3 Customization using plugins

Plugins allow you to customize the extraction. Mainly you can transform the field and line item contents based on the specific invoice. You can also access the text data page wise and extract invoice specific data.

To write a plugin, you need to write a new python file and drop it under src/plugin. You should specify the name of the plugin in the invoice template file.

Template	<pre> "line_end": "Total" }, "plugin": "flipkart-tech-connect" }</pre>
Plugin Python	<pre> #!/usr/bin/env python def transform(_type, type_name, type_value): print _type, ' ', type_name, ' ', type_value return type_value def transform_line_item(index, type_name, type_value): print index, ' ', type_name, ' ', type_value return ' ' + type_value + ' ' def post_processor(extracted_data, container_instance): print len(container_instance.pagewise_rows_of_y0_textboxes) return extracted_data</pre>

Three methods need to be implemented

transform	Method is called for every field. type_name is the name of the field type_value is the extracted value of the field name Either the transformed value or the original value needs to be returned.
transform_line_item	Method is called for every field. type_name is the name of the column type_value is the extracted value of the column name Either the transformed value or the original value needs to be returned.
post_process or	Refer ObjectLayoutContainer.py for the methods that provide you access to the text boxes and their data.

