Automatic Extraction of References Between Text and Charts

Sherry Ruan & Dae Hyun Kim
May 5, 2016
Introduction

- Analyze how information is referenced at scale
- Automatically extract references between text and charts
- Use extracted references to build an intelligent interactive paper reading tool (End goal)
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
- Extracting References via Crowdsourcing [Kong et al. 2014]
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
  
  ✓ Identify the chart type using computer vision and machine learning

- Extracting References via Crowdsourcing [Kong et al. 2014]
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
  - Identify the chart type using computer vision and machine learning
  - Extract the graphical marks and infer the underlying data

- Extracting References via Crowdsourcing [Kong et al. 2014]
Related Work

● Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
  ✓ Identify the chart type using computer vision and machine learning
  ✓ Extract the graphical marks and infer the underlying data
  ✓ Apply cognitive design principles to improve graphical perception

● Extracting References via Crowdsourcing [Kong et al. 2014]
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
  - ✔ Identify the chart type using computer vision and machine learning
  - ✔ Extract the graphical marks and infer the underlying data
  - ✔ Apply cognitive design principles to improve graphical perception
  - ☐ Cannot relate chart data to text data

- Extracting References via Crowdsourcing [Kong et al. 2014]
Related Work

● Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]

● Extracting References via Crowdsourcing [Kong et al. 2014]

✓ Present a crowdsourcing pipeline to extract the references
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
- Extracting References via Crowdsourcing [Kong et al. 2014]
  - Present a crowdsourcing pipeline to extract the references
  - Provide an interactive document viewing application
Related Work

- Revision: Automated Classification, Analysis and Redesign of Chart Images [Savva et al. 2011]
- Extracting References via Crowdsourcing [Kong et al. 2014]
  - Present a crowdsourcing pipeline to extract the references
  - Provide an interactive document viewing application
  - Crowd algorithm is not scalable
End Goal
Our Current Annotation Approach
Plans & Current Progress

Current

UI for Annotation

Data Collection (Crowdsourcing)

Annotation Automation

Project Merging & Verification

PDF -> HTML (Table Extraction)

PDF -> HTML Automation

HTML -> PDF (+Interactivity)
Plans & Current Progress

UI for Annotation

Data Collection (Crowdsourcing)

Annotation Automation

Project Merging & Verification

PDF -> HTML (Table Extraction)
PDF -> HTML Automation

HTML -> PDF (+Interactivity)
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
- Non-commercial
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
  - ABBYY
  - Adobe Acrobat Reader
- Non-commercial
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
  - ABBYY
  - Adobe Acrobat Reader
- Non-commercial
ABBYY - Description

- Russian firm specializing on optical character recognition (OCR)
- ABBYY FineReader provides exporting PDF files to HTML format
- How it works
  1. Transform PDF into image
  2. OCR + internal technology
Your Basic Bill Plan
Your Bill Plan (as on 27-Jan-2010):
Get More Plan Bill Plan Charges Including Rental Rs. 199
Incoming - Local Free
To Airtel To Other Mobile To Landline To Airtel CUG
Outgoing - STD (Rs./Min) 0.3 1 1 0.3
Outgoing - ISD (Rs./Min) 1.5 1.5 1.5 1.5
SMS Local (Rs./SMS) 1
SMS National (Rs./SMS) 1.5 1.5 1.5
Roaming (Rs./Min) 6.40/9.20/10.00/11.00/15.00/40.00/45.00/100.00/550.00/1000.00
Benefits of tariff discounts through additional packages if any are reflected in discounts section

<table>
<thead>
<tr>
<th>1. One Time Charges</th>
<th>Description</th>
<th>From Date</th>
<th>To Date</th>
<th>Amount</th>
<th>Discoun Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Bill Period</td>
<td>18/01/2010</td>
<td></td>
<td></td>
<td>75.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Late Fee</td>
<td></td>
<td></td>
<td></td>
<td>75.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>75.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Two Period Charges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>From Date</td>
</tr>
<tr>
<td>To Date</td>
</tr>
<tr>
<td>Amount</td>
</tr>
<tr>
<td>Discount Units</td>
</tr>
</tbody>
</table>
ABBYY - Strengths & Weaknesses

Strengths

- Works even for image tables
- Able to deal with multiple table types (e.g., joined rows, columns)
- Marks portions with low certainty

Weaknesses

- Not a freeware
- Performs OCR on even text-based PDF (additional source of inaccuracy)
- Cannot properly handle LaTeX-generated text
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
  - ABBYY
  - Adobe Acrobat Reader
- Non-commercial
Adobe Acrobat Reader - Description

- PDF standard set by Adobe (1993)
  - Some technologies defined only by Adobe
- Exporting PDF files to HTML format provided as function of Adobe Acrobat Reader Pro
- Not much known about internal logics
Adobe Acrobat Reader - Examples

Adobe

study to the end as well as the length of the text that they copied at one time. The large standard deviations here are likely because three participants read very little or not at all, only using the Alphabet Keyboard to enter text, while some read letters throughout the study. Interestingly, participants who read letters less tended to re-read shorter text at the end, while those who read letters more tended to re-read shorter text.

Table 6. Mean and median of count of words taken to reach a target letter for each 2-day temporal bin.

Table 7. Mean and median of letter counts and length of text read at one time, per 2-day temporal bin.

Our analysis of accuracy rates shows that the average CSS error rate across all participants and across all techniques was 15%, meaning that allowing for multiple spaces between words only introduced an error of 2% into the data. Regarding the Error Rate (i.e., corrected errors excluding destructive deletions), we found that there was a significant effect of Technique (F(4,399) = 5.007, p < 0.001) and Text (F(4,399) = 12.16, p < 0.001). There was no significant effect of Technique*Text (F(16,1596) = 0.77, p > 0.47) indicating that the Error Rate improvement over time was similar for all techniques. Figures 7 shows the largest mean squares Error Rate of each of the techniques within each bin indicating that both the Alphabet and Triple Keyboard incurred a generally lower Error Rate than the Scrolling, Triple Scrolling, and Rubime techniques (although the only significant difference in overall error rate was between Alphabet, Scrolling, and Scrolling, see Table 4).

Table 8. Overall least square mean difference in Error Rate between techniques. Statistically significant differences (p<.05) are starred.

The final survey data showed that the majority of students in each group would choose to use Scrolling and Triple Scrolling if they had to use one technique regularly in their classes (57.1% for Scrolling and 71.4% for the Triple Scrolling). Table 9 summarizes the survey results.

Table 9. Percentage of students choosing each technique to answer the corresponding survey question.

Discussion

Alphabet, Triple Keyboard, and Reuse were comparatively fast to learn and use after six days of practice, and were all faster than the scrolling techniques. However, students demonstrated a relatively higher error rate when using the Reuse technique compared to Alphabet and Triple Keyboard, which both caused fewer errors than the scrolling techniques.

Interestingly, although the scrolling techniques (Scrolling and Triple Scrolling) were significantly slower than the other techniques and produced more errors, participants tended to favor these techniques the most. From our observation, this could be due to several factors. First, scrolling techniques present less of an acquisition problem because students can scroll and slide through every character in the text, allowing the scrolling technique to be a less complex technique to learn and use. This makes it easier to ignore other text quickly. Third, all students tested in our study were able to fully control the mouse and quickly navigate the text boxes, while holding the mouse down with their other hand. This holding style is also difficult for the non-scrolling techniques because they require targeted movements.

Table 10. Comparison of favorite techniques and design factors (including cost, if each of our techniques was based around a single mouse per student). Figure 6 visualizes these factors using bar charts to show the relative preference for each factor.
Adobe Acrobat Reader - Strengths & Weaknesses

Strengths

• Translation format has very high accuracy
• Able to deal with multiple table types (e.g., joined rows, columns)
• Preserves table style

Weaknesses

• Not a freeware
• Cannot properly handle LaTeX-generated text
Issues of Using Commercial Softwares

- Licensing issues
- Internal logics hidden
- Difficult or impossible to tweak to certain purposes
- Throws away all other information
  - Will require use of another pdf extraction tool for converting annotations on HTML back onto PDF

-> Use commercial softwares for gathering a small initial data set.

& Use open-source projects for automation
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
- Non-commercial
  - Tabula
  - pdffigures
  - pdf2htmlEX
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
- Non-commercial
  - Tabula
  - pdffigures
  - pdf2htmlEX
Tabula - Description

- Open-source project written in Java
- Based on Apache PDFBox
- Designed specifically for table extraction
- UI available
Tabula - Internal Mechanism

- Word / Paragraph Detection (Character Clustering)
- Line Detection (Hough Transform)
- Boundary Detection
- Column / Row Detection (Using Separators / Word Clusters)
**Tabula - Examples**

<table>
<thead>
<tr>
<th>Technique</th>
<th>Screen Space Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAG II</td>
<td>17.6%</td>
</tr>
<tr>
<td>Alphabet, Fish-Eye, Collaborative Blanks, Reuse</td>
<td>15.9%</td>
</tr>
<tr>
<td>Triplet Keyboard</td>
<td>12.8%</td>
</tr>
<tr>
<td>Multi-tap</td>
<td>11.5%</td>
</tr>
<tr>
<td>Quintuplet Keyboard</td>
<td>10.3%</td>
</tr>
<tr>
<td>Quintuplet Scroll</td>
<td>1.5*N%</td>
</tr>
<tr>
<td>Triplet Scroll</td>
<td>0.8*N%</td>
</tr>
<tr>
<td>Scroll</td>
<td>0.3*N%</td>
</tr>
<tr>
<td>Morse Code, EdgeWrite</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1. Percentage of screen space used by each technique. 

\[ N \text{ represents the number of users.} \]

\[ y = cx^k \]

<table>
<thead>
<tr>
<th>Technique</th>
<th>[ y = 2.6x^{-15} ]</th>
<th>[ R^2 ]</th>
<th>ANOVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td></td>
<td>.19</td>
<td>( F_{1,139} = 73.19, p &lt; .0001 )</td>
</tr>
<tr>
<td>Triplet Scroll</td>
<td></td>
<td>.31</td>
<td>( F_{1,139} = 60.92, p &lt; .0001 )</td>
</tr>
<tr>
<td>Triplet Keyboard</td>
<td></td>
<td>.40</td>
<td>( F_{1,180} = 120.45, p &lt; .0001 )</td>
</tr>
<tr>
<td>Scroll</td>
<td></td>
<td>.14</td>
<td>( F_{1,95} = 16.07, p &lt; .0001 )</td>
</tr>
<tr>
<td>Reuse</td>
<td></td>
<td>.18</td>
<td>( F_{1,102} = 21.86, p &lt; .0001 )</td>
</tr>
</tbody>
</table>

Table 3. Power curve, \( R^2 \) value, and ANOVA results for each text entry technique.
Tabula - Strengths & Weaknesses

**Strengths**

- Open-source
- Specifically designed for table detection & data extraction

**Weaknesses**

- A lot of false detections
- A lot of misses
- Cannot deal with joined rows / columns
- Cannot properly handle LaTeX-generated text
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
- Non-commercial
  - Tabula
  - pdffigures
  - pdf2htmlEX
pdffigures - Description

- Created by Allen Institute for Artificial Intelligence
- C. Clark, S. Divvala, Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers, AAAI 2015
- Written in C++
- Based on Poppler
- Designed for figure / table extraction with associated captions from computer science papers
pdffigures - Internal Mechanism

Caption Identification Using Key Words (Table, Figure, etc.)

Region Identification Using ‘Lack of Text’

Caption Assignment Using Adjacency
pdffigures - Strengths & Weaknesses

**Strengths**
- Open-source
- Performs well on figure / table detection

**Weaknesses**
- Designed for figure / table detection, NOT data extraction
- Does not distinguish figures vs. tables
- Designed specifically for CS papers
- Some apparent mis-detection / bad bounding box patterns (e.g., title page)
Tools for PDF to HTML Conversion / Table Extraction

- Commercial
- Non-commercial
  - Tabula
  - pdffigures
  - pdf2htmlEX
**pdf2htmlex - Description**

- Open source project written in C++
- Based on Poppler
- Designed for accurately converting PDF into HTML
- How it deals with tables
Examples of Computer Science Class Notes

Alberto Gómez (UIO) - Kalipada: A novel-based technique where text is written by traversing edges and diagonals of squares instead of by physical keys. We modified this to allow students to traverse squares with screen edges.

Table 1: Percentage of screen space used by each technique.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Screen Space Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td>17.5%</td>
</tr>
<tr>
<td>Collectedist</td>
<td>12.5%</td>
</tr>
<tr>
<td>Keyboard (left)</td>
<td>15.3%</td>
</tr>
<tr>
<td>Smartphone</td>
<td>13.7%</td>
</tr>
<tr>
<td>Tablet</td>
<td>8.3%</td>
</tr>
</tbody>
</table>

**Evaluation:**
We conducted a series of evaluations to determine the most effective on-screen technique with respect to the design considerations described earlier. Given the large number of techniques, we first conducted two preliminary evaluations to rule some before going on to a formal study of the five most promising techniques.

All our studies use the same single-blind, randomized, post-test design. While only some of the techniques are ethically tested without all of them being used for this evaluation, each of these studies has been appropriately evaluated in a singled-out scenario.

**Acknowledgments:** We are grateful to the students who participated in this study. We thank the Children's Hospital Trust (CHT), a non-profit organization in Bangladesh, for providing an excellent educational environment. We gratefully acknowledge and thank the primary and secondary school children on whose evaluative data are partially based.

**References:**
pdf2htmlEX - Strengths & Weaknesses

Strengths

- Open-source
- Performs exceptionally well for PDF to HTML conversion
- Handles LaTeX-generated text well
- Keeps the original styling of tables

Weaknesses

- Does not perform any table detection
Key Challenges

- Have an **accurate** system that **generalizes** to a reasonably wide range of PDF’s and tables
- Find **suitable coding basis** so that I won’t be ‘reinventing the wheel’
Failed Attempts

- Open source library use with commercial software basis
  - Initial conversion using a commercial software + customization of HTML using an open source library
  - Adobe Acrobat Reader + Tabula to recover table location data in the original PDF
  - Limited success -- okay as a prototype
  - Issues:
    - Does not fundamentally resolve all issues of using commercial software
    - The weaknesses of both add up
Current Attempts

● Building on top of an open source PDF-handling library
  ○ Building on top of PDFMiner (PDF extraction tool)
    ■ Requires solid understanding of PDF format
    ■ Will need a custom-written table detecting / parsing heuristics
      ● Written in Python - compatibility with already discussed libraries difficult
      ● Could end up just ‘reinventing the wheel’ (PDF extraction tool -> table extraction / detection)
  ○ Building on top of pdffigures + pdf2htmlEX (both C++)
    ■ Figure / Table detection is granted
    ■ Access to accurate text extraction
    ■ Will need to do:
      ● Figure / table distinction
      ● Row / column detection of tables
Plans & Current Progress

Current

UI for Annotation

Data Collection (Crowdsourcing)

Annotation Automation

Project Merging & Verification

PDF -> HTML (Table Extraction)

PDF -> HTML Automation

HTML -> PDF (+Interactivity)
Data Collection Using Crowdsourcing

After automatically extracting html tables from pdfs ......

- Pair each extracted table with its corresponding paragraphs
- Automatically generate easy-to-use websites for reference annotation
- Run pilot studies to collect and analyze different users’ annotation results
- Use crowdsourcing to collect a large amount of annotated data
UI for Annotation: Design

- A single-page interface corresponds to a paragraph-table pair
- Currently focus on text-cell relationships
- Allow multiple text/cell selections in one reference at the same time
- Display an intermediate representation of selections immediately after the user's selection
- Provide delete buttons to remove unsatisfying references
- Highlight a selected reference in the interface on mouse hover
- Build a mapping system between saved information and html interface
- Can upload and review saved results
- Click elsewhere unselects everything
UI for Annotation: Information Storage

- Both content and location information of text and cells is saved
- Contents are saved as plain strings
- Text locations are saved in terms of character offset and text length
- Cell locations are saved in terms of row index and column index
- Information is saved as CSV now; Will change to JSON in future
UI for Annotation: Demo

RGB-W: When Vision Meets Wireless

Table 3: Performance of assigning the correct mac ID to an individual. The number of people indicates the number of people in a scene assuming all people are broadcasting W data.

<table>
<thead>
<tr>
<th>Number of People</th>
<th>Greedy</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>61.7%</td>
<td>64.0%</td>
</tr>
<tr>
<td>4</td>
<td>52.0%</td>
<td>57.2%</td>
</tr>
<tr>
<td>6</td>
<td>45.6%</td>
<td>53.4%</td>
</tr>
<tr>
<td>8</td>
<td>36.1%</td>
<td>45.3%</td>
</tr>
<tr>
<td>10</td>
<td>27.3%</td>
<td>30.2%</td>
</tr>
<tr>
<td>12</td>
<td>21.0%</td>
<td>28.6%</td>
</tr>
</tbody>
</table>

The RGB-W data enables the use of a new similarity measure to solve the tracking problem. We study the performance of assigning the mac IDs to detected individuals given their rough localization. Table 3 presents the performance of the assignment as a function of the number of individuals in the scene. During the experiments, individuals were moving in highly dense manner, i.e., 1 to 2 meters away from each other even when two individuals were present. The assignment is based on minimizing the global distances between the detections from RGB-W and W only. In Table 3, we can see that our proposed method is outperforming the greedy approach but is still challenging. The success rate is not high. Future work can investigate on how to increase the performance of such task by comparing the temporal dynamics of the W with respect to the tracklets.

Upload your annotated results

Choose File  No file chosen  Submit  Save

Text Selected: 30.2%

References:

- Text: "study", "experiments"  Cells: [4, "52.0%", "45.6%"]  Delete
- Text: "greedy"  Cells: [10, "27.3%"]  Delete
Annotation Automation: First Attempt

- Baseline algorithm uses exact word match
- Attempt different preprocessing methods: normalizing, stemming and lemmatization
- Results largely depend on papers given
Current Challenges

- Implement tools that can extract complex tables (i.e., spanning multiple rows, without borders)
- Find a suitable balance between taking what the existing libraries offer and what we implement
- Analyze semantic meanings of words to associate them with data in tables
- Generalize to different types of charts
Next Steps

- Work on automating table extraction (PDF -> HTML)
- Improve the interface so that it displays extracted tables in the same style as in pdf files
- Run pilot studies on a small group of people to analyze how people reference information
- Collect a large amount of annotated data using our interface and crowdsourcing
Questions?