Cross Domain Assessment of Document to HTML Conversion Tools to Quantify Text and Structural Loss During Document Analysis

Kyle Goslin
Department of Informatics and Engineering,
Institute of Technology Blanchardstown,
Blanchardstown Road North,
Dublin 15, Ireland
kylegoslin@gmail.com

Markus Hofmann
Department of Informatics and Engineering,
Institute of Technology Blanchardstown,
Blanchardstown Road North,
Dublin 15, Ireland
markus.hofmann@itb.ie

***Abstract*—During forensic text analysis, the automation of the process is key when working with large quantities of documents. As documents often come in a wide variety of different file types, this creates the need for tailored tools to be developed to analyze each document type to correctly identify and extract text elements for analysis without loss.**

These text extraction tools often omit sections of text that are unreadable from documents leaving drastic inconsistencies during the forensic text analysis process. As a solution to this a single output format, HTML, was chosen as a unified analysis format. Document to HTML/CSS extraction tools each with varying techniques to convert common document formats to rich HTML/CSS counterparts were tested. This approach can reduce the amount of analysis tools needed during forensic text analysis by utilizing a single document format.

Two tests were designed, a 10 point document overview test and a 48 point detailed document analysis test to assess and quantify the level of loss, rate of error and overall quality of outputted HTML structures.

This study concluded that tools that utilize a number of different approaches and have an understanding of the document structure yield the best results with the least amount of loss.

I. Introduction

In a number of different sectors, large repositories of documents are often built up each with a wide variety of different formatting techniques and styles by a number of different authors. When a forensic text analysis of the documents in these repositories is needed, a manual analysis becomes no longer feasible.

Although various different file types are used, the vast majority are often a sub collection of these file types. These file types include Microsoft .doc, .docx, .ppt, .pptx and the now open standard .pdf. Each document contains different internal representations such as plain text, XML and binary with different approaches used during the document rendering process.

This variety of different file types provides a fundamental problem during the process of forensic text analysis as a variety of different tools would need to be created to deal with each different file format.

As a solution to the multi-filetype problem, a single style and representation-based format, such as HTML can be used as a bridging format to which all documents can be converted into. As HTML has been around for a number of years a multiple of tools currently exist for converting from common file formats into HTML. HTML when utilized correctly can be applied to create identical representations of the original documents and used in place of the original document to provide better searchability and more flexibility when analyzing documents. The quantity of files in a repository can also cause an issue as manually converting files no longer becomes a feasible option due to the time required to convert each file. For this reason, an automated conversion tool is needed.

This paper outlines the background issues that arise during the conversion process of documents to HTML for forensic text analysis in Section II. Section III outlines the document types and content variations that were dealt with during this study. Section IV outlines the various tools and approaches that are currently available for converting documents to HTML/CSS counterparts.

Section V outlines Experiment 1 and Experiment 2 that were used to gauge the level of loss and quality of outputs for the selected tools. Finally Section VI reflects on the overall findings from this study.

II. Background

Converting PDF files to HTML can be done with a number of different tools, each of which implement varying approaches [1][2]. HTML documents are often generated to create web based representations of previously unindexable documents as they are fully text based. This need to extract text from documents has led to a number of assessments on the PDF to HTML conversion process being done in this area [3].

The utilization of document layout and styling information has been underway for a number of years. Segmentation based upon HTML structure [4] has been used in the process of information extraction. The utilization of DOM tree and bounding boxes to aid additional processing has been used for a number of different purposes such as aiding search and text matching [5], [6], [7].

978-0-7695-5062-6/13 $26.00 © 2013 IEEE
DOI 10.1109/EISIC.2013.22

Each tool for converting to HTML can contain any number of different flaws during the generation process when footnotes, table sizes and non-standard formatting techniques are used. This poses the initial problem of the quality of the output text being misaligned and sections of text being omitted along with words being spelt wrong due to bad OCR identification [8].

These uncertainties in output quality created the need for output quality to be correctly gauged preventing these issues from affecting future forensic text analysis steps.

III. Document Analysis

Each document consists of a number of different pages or slides, each with a varying amount of content in different layouts utilizing different formatting. To aid the process of classification, all text in the document was divided into separate categories to describe the type of text.

- Text with a combination of large text, styling and positioning typically at the top of the page with the intention of catching the attention of the reader first is considered the main heading of the page if no HTML H1 tag or similar exists.
- Text with an applied emphasis above the main body of text, typically paragraph headings or section identifiers is considered a level 2 heading.
- A large body of text with a common size used throughout the page/size with no implied emphasis and smaller than any other content deemed level 1 or level 2 is considered the body text.

Emphasis of the text can be applied in a number of different ways. With heading and sub-headings, the size of the text is one of the most common ways to represent emphasis. Other more subtle alternatives such as bold, underline and italics can often be seen. Paragraphs and bodies of text the emphasis of a piece of text is often done through the use of variations in font style and not just through the use of text size.

A. Data Extraction Loss

During the extraction process a quantity of data is often lost. Unlike working with simple text files, text editing tools offer a variety of different styling and content structuring tools. The result of using these tools is rich styled content. The main downfall of using these tools is that previously simple text is often represented in non-text based elements e.g. complex font and styling of title text is represented as an image.

During the text extraction process, if additional text recognition tools are not included the text represented as images is often lost. This lost text can contain level 1 headings and useful descriptive text leaving the document as a quantity of body text without any headings.

Documents also often include a number of large images created with other software packages such as charts or graphs. These graphs often contain additional titles and descriptive body text that is essential to the document when considered as a complete body of text. If the bodies of documents are mainly image based and if the extraction process is not accustomed to identifying text in the images, this text is often lost.

IV. Extraction Approaches and Available Tools

Extracting data from documents can be done in a number of different ways, each with a varying amount of emphasis on different aspects of the document and text. These approaches can be summarized in the following ways:

- Higher accuracy of the text and less focus on the positioning and layout of the text
- Higher emphasis of the positioning of the text with less emphasis on quality of text
- Higher emphasis on the formatting quality of the text e.g. bold, italic, underline

First an assessment was done on the different approaches that could be taken. A *pure Optical Character Recognition (OCR)* approach views the document as a single picture and attempts to extract the text from the image. A common output of OCR tools is hOCR [9], a hybrid HTML format that contains formatting information and the identified text.

Document Layout Analysis (DLA) such as **Ocropus**[1] and **OCR Feeder**[2] combine both OCR engines and additional document layout knowledge to make the best attempt at converting the input file into a HTML output. These tools work specially well with documents with known standard layouts, such as conference and journal papers whereby the header and descriptive information is always in predefined location. This static location provides the tool and understanding of the document before attempting to identify and extract the content.

A number of *conversion libraries* exist that attempt to make an understanding of the document types to create a better output. Each of these tools utilizes a number of different techniques to produce the desired output. **pdftohtml**[3] has been a widely used tool for converting PDF files to their HTML counterpart. **unoconv**[4] is a set of bindings for the popular open source document suite OpenOffice (recently forked LibreOffice). **Adobe Acrobat Pro**[5] is a desktop application which offers PDF files to be converted into a number of different desired output file formats. When unsure of a piece of text, additional OCR tools are automatically run by the tool.

V. Quality Assessment

In order to test the number of tools that exist for conversion to HTML, a number of different tests were created to test each different tool correctly. **Experiment 1** (Section V-A) is designed to test the functionality overview of all tools. **Experiment 2** (Section V-B) is a complete test designed to test a smaller number of tools in more detail. In this test the finer document features and quality of the selected tools was assessed. This section describes these tests and how each tool performed.

[1] http://code.google.com/p/ocropus/
[2] https://live.gnome.org/OCRFeeder
[3] http://pdftohtml.sourceforge.net/
[4] http://dag.wieers.com/home-made/unoconv/
[5] http://www.adobe.com/products/acrobatpro.html

A. Exp 1: Output Quality Overview

Table I outlines the tools which were tested. Each tool was tested under 10 different general output analysis checks (seen in Table III) such as the quality of the overall output, font colour analysis, font size errors and omitted text. The tool type is identified in Table I as either Pure OCR, DLA or Combined (uses document understanding and OCR approaches during processing).

TABLE I. EXPERIMENT 1 TESTED TOOLS

Identifier	Tool Name	Tool Type
T1	Tesseract OCR 3.02	Pure OCR
T2	Ocropus 0.6	DLA
T3	pdftohtml 0.40a	Combined
T4	LibreOffice 3.6	Combined
T5	unoconv 0.5(LibreOffice 3.6)	Combined
T6	Adobe Acrobat Pro 11.0.0	Combined

1) Data Set Gathering and Sample Set: Document repositories consist of a wide range of documents, from a variety of different sources. To replicate this and to remove any bias from the data collection process, a random data collection was done using Google with keywords and file type specific requirements. An example of this is for engineering related documents, a search was done for *engineering filetype:pptx*. The results that were returned could then be classified as random as they came from a variety if different sources. For each test the data set consisted of 5 different topic domains (Computer Science, Engineering, Medicine, Social Care, Psychology), 5 different file types (doc, docx, pdf, ppt, pptx) in each of which 3 random slides were then selected to be tested. This created a total of 75 slides to be tested for each tool in a number of different areas.

2) Test Results: Bold, Italics and Underline of the text was assessed to see if it was correctly represented in the output. Font variations was used to identify if differences in the original document fonts were correctly represented in the output. As a number of tools did not create HTML/CSS output of an acceptable standard, an assessment was done to identify if the outputted HTML structures e.g. lists and tables were outputted in the relative HTML structures and not just as visually acceptable HTML layout.

In some cases tools failed to produce any output for specific document types preventing slides/pages to be extracted for analysis. These failed conversions are identified in Table II.

TABLE II. NUMBER OF FAILED CONVERSIONS OF PAGE/SLIDES

Identifier	Tool Name	Number of Failed Conversions
T1	Tesseract OCR 3.02	3 / 75
T2	Ocropus 0.6	37 / 75
T3	pdftohtml 0.40a	3 / 75
T4	LibreOffice 3.6	15 / 75
T5	unoconv 0.5(LibreOffice 3.6)	15 / 75
T6	Adobe Acrobat Pro 11.0.0	4 / 75

T1 Tesseract is a pure OCR solution. To use this tool documents are first converted into image based representation. Overall the results of the tool were unsatisfactory as very little emphasis on the formatting and the styling of the text was accounted for. A number of different errors were also encountered in the text that was produced. These errors are generally attributed to the complexity in the styling of the text. Table III outlines the results of this test. Additional checks were done to ensure that the output was an accurate representation of the original document and the outputted HTML could be deemed *detailed* e.g. the HTML/CSS was not a simplified output of the original document.

TABLE III. EXP 1 - T1: TESSERACT

	T1 - *Yes*	**T1 -** *No*	**T1 -** *N.A.*
Bold Text	1%	57%	42%
Italic Text	0%	35%	65%
Text Bleed	4%	57%	39%
Font variations	5%	85%	10%
Correct HTML	0%	100%	-
Paragraph Sentence grouping	79%	21%	-
Correctly represent original	30%	70%	-
Detailed formatting	0%	100%	-
Text Errors	40%	60%	0%
Original Text Retained	76%	24%	0%

T2 Ocropus required all slides/pages to be converted into image based representations before process. This tool did not correctly convert the files having 37 out of 75 unsuccessful conversions. Most of the fails are attributed to high style based formats, e.g. .ppt and .pptx presentations and had more successful conversions with .doc and .docx files. Although successful, the overall quality of the output was bad having a high error rate. The styling of the output was also quite sparse. Due to the high number of fails during the conversion process this tool was automatically excluded from any further analysis.

T3 pdftohtml requires all documents to be converted into their PDF counterpart. To do this, native file type conversion tools were used. All documents were successfully converted into HTML. This tool focuses mainly on the visual layout of the document to create a high, visually accurate layout representation. The main issue with this process is that the HTML that is created was a complex representation of HTML div tags. Sentences and natural groupings of text were broken down into smaller elements breaking the semantics of the text. The representation of HTML structures such as bullet lists and tables were replaced in the documents in favor of the div based representations. A number of different flaws were seen in the outputted documents such as colour bleedings between words and variations in fonts not being correctly represented. In a number of cases, errors were also present in the text whereby two letters that were close together such as *t* and *i* was misrepresented as an @ sign. The font in this tool is often defaulted to Times New Roman. Underlines are inconsistent throughout documents. Overall the focus of the tool was placed upon the visual layout of the text and not the quality of the text or the HTML. Table IV outlines the results.

TABLE IV. EXP 1 - T3: PDFTOHTML

	T3 - *Yes*	**T3 -** *No*	**T3 -** *N.A.*
Bold Text	36%	25%	39%
Italic Text	11%	20%	69%
Text Bleed	43%	11%	46%
Font variations	99%	1%	0%
Correct HTML	0%	100%	-
Paragraph Sentence grouping	100%	0%	-
Correctly represent original	97%	3%	-
Detailed formatting	100%	0%	-
Text Errors	3%	97%	0%
Original Text Retained	100%	0%	0%

T4 LibreOffice was chosen as it had native exporting ca-

pabilities. The initial attempts of converting straight to HTML were not successful. The tool was only able to convert 15 out of 75 sides/pages correctly. Of the successful conversions only .doc files were successful. All other file types were not successfully converted. As a large portion of the conversions were not possible, this tool was excluded from any further analysis.

T5 unoconv provided an additional set of bindings to the LibreOffice. During the conversion process bridging is done between other file types to allow files to be successfully converted to HTML. This tool was successful in converting 60 of 75 pages/sides. All file types in exception to PDF files were successfully converted. The output that was created lacked style and font variation. All representations were trimmed down removing any additional images and text. No attempt was made at converting image based text into regular text. Numbered lists were often represented as regular bullet lists. Although the bullet lists were structured in correct HTML it was often the case that text above or below the list would become another item in the list. Title text was often omitted from the documents, pulling a level two header higher up to be a level one heading. Image based slides were completely ignored by the tools, creating a number of blank pages/slides to appear throughout the documents. All footers and headers were routinely excluded from the process, printing only once at the start and once at the very end of the document. This overview representation of the documents caused a large quantity of text to be omitted. Sequences of full stops and lines were often excluded from the output. Two column sentences are often merged together creating one sentence, with in between white space omitted. Table V outlines the results.

TABLE V. EXP 1 - T5: UNOCONV

	T5 - *Yes*	**T5 -** *No*	**T5 -** *N.A.*
Bold Text	40%	5%	55%
Italic Text	25%	3%	72%
Text Bleed	8%	30%	62%
Font variations	82%	5%	13%
Correct HTML	36%	64%	-
Paragraph Sentence grouping	87%	13%	-
Correctly represent original	60%	40%	-
Detailed formatting	92%	8%	-
Text Errors	7%	93%	0%
Original Text Retained	77%	23%	0%

T6 Adobe Acrobat Pro allows the conversion from any file type to html. For documents such as doc, docx, ppt and pptx a conversion is done through Acrobat using native conversion tools e.g. Microsoft Office to convert from the original file type to PDF. The resulting HTML that was exported from the tool proved to be of very high quality. The formatting included in the output was detailed HTML. Two column layouts in some cases have been merged breaking semantics. Sentence and paragraph representations however are very good, having little to no break down. The detail of the text formatting creates complex representations of the text. This causes the text to run in a diagonal formation, although all text on a per-page/per-slide basis is consistent and correctly positioned relative to that page/slide. Styling information from the original documents was correctly retained. Font emphasis such as underline, bold and italic was correctly represented. Little to no bleed across words was found for any of the styling of text. Table VI outlines the results.

TABLE VI. EXP 1 - T6: ADOBE ACROBAT PRO

	T6 - *Yes*	**T6 -** *No*	**T6 -** *N.A.*
Bold Text	51%	0%	49%
Italic Text	28%	2%	70%
Text Bleed	59%	3%	38%
Font variations	89%	1%	10%
Correct HTML	100%	0%	-
Paragraph Sentence grouping	91%	9%	-
Correctly represent original	73%	27%	-
Detailed formatting	100%	0%	-
Text Errors	0%	100%	0%
Original Text Retained	100%	0%	0%

3) Results Review: Overall a combined approach to the text extraction proved to be the best solution. OCR/DLA based solutions did not provide the detailed HTML that was needed and all font styling was generally omitted or wrongly represented. Adobe Acrobat Pro proved to give the best results and the highest accuracy when considering font style and detailed HTML output quality. unoconv provided a good quality output, but was progressively worse as the complexity of the documents increased and did not handle PDF files.

For non-layout based tools, two column layouts always cause issues with the merging of sentence breaking semantics. Text at non-standard orientations such as text at a 40 degree angle caused issues. From the quality of the outputs, pdftohtml, unoconv and Adobe Acrobat Pro were selected for detailed further analysis.

B. Exp 2: Complete Output Testing

From the original set of tools, a smaller set was then considered for a more detailed analysis of output quality. Table VII outlines the selected tools. A test was developed to assess 48 different aspects of each document to provide a better understanding of output quality, content loss, layout and style of the document.

TABLE VII. EXP 2 DETAILED TOOL TESTING

Tool Name	**Tool Type**
pdftohtml 0.40a	Combined
unoconv 0.5(LibeOffice 3.6)	Combined
Adobe Acrobat Pro 11.0.0	Combined

1) Data Set Gathering: When gathering the data set for the second test, a similar approach was taken to the first test. A Google search was performed across 5 domains (Computer Science, Engineering, Medicine, Social Care, Psychology), of these domains, 5 files types were selected (doc, docx, pdf, ppt, pttx) and 10 of each were then selected. This created a data set of 250 files that each had a variety of different layouts, style and content.

2) Results: This section outlines the results of how the tools performed against each other.

Overall loss To correctly identify the amount of loss during the extraction process a metric was created to allow the loss of data to be correctly quantified relative to the type of data that was lost. Table VIII outlines the categories of loss between 0 and 5. Although any loss in the data is not desirable, a certain quantity of data may be lost during the extraction process.

TABLE VIII. CATEGORIES OF LOSS

Category	Description
0	No Loss
1	Very Minimal Body Text Loss
2	Small Sub-headings Lost
3	Very Limited Image Based Content Loss
4	Image Based Content Loss
5	H1 Heading Lost

pdftohtml performed very well during the extraction process, with 83% document correctly converted without any loss. Behind this was the Adobe Acrobat with 81% successful conversions. Last came unoconv with 52%. The lower than usual score for this tool was attributed to the fact that it did not support any PDF conversions. This automatically excluded a large data set from the process.

For the loss of headings from pages/slides which is considered valuable (category 5), unoconv score poorly overall scoring 12 complete failures during the conversion process. pdftohtml had 11, unoconv had 9 and Adobe had 15. These were generally a level playing field for the amount of loss. Overall The highest quantity of loss per file type was on pptx files. With an an approximate 4 time increase in the number of scored loss issues. Level 3 and 4 combined put Adobe ahead with only 23, pdftohtml scored 25 and unocov scored 36.

Font Analysis The size, style and colour of fonts in the document to be correctly retained was of great importance as they are the fundamental metrics for indicating the emphasis on text. Table IX outlines the results showing that the retention of font size was best by Adobe Acrobat. The high score for pdftohtml can be attributed to default Times New Roman fonts being accepted as the correct font.

TABLE IX. FONT SIZE RETENTION

Tool	Yes	No
pdftohtml	99%	1%
unoconv	91%	9%
adobe	100%	0%

Retaining the font style can provide an insight into the changing of emphasis on text. Some slight variations in font style can highlight this emphasis. Table X outlines the results of this test, once again putting the Adobe tool ahead.

TABLE X. FONT STYLE RETENTION

Tool	Yes	No
pdftohtml	31%	69%
unoconv	49%	51%
adobe	92%	8%

The retention of font color was a problem that was originally highlighted in the pdftohtml tool. For this reason, a check was done across all tools. unoconv output had a large quantity of black text, which was the default colour. This affected the final grade for unoconv making it slightly higher than usual. Table XI outlines these results.

TABLE XI. FONT COLOUR RETENTION

Tool	Yes	No
pdftohtml	96%	4%
unoconv	57%	43%
adobe	98%	2%

pdftohtml originally highlighted issues of colour bleed, whereby text beside the coloured text becomes that colour also. Table XII shows that bleed was most prominent in pdftohtml.

TABLE XII. FONT COLOUR BLEED

Tool	Yes	No
pdftohtml	33%	67%
unoconv	0%	100%
adobe	2%	98%

Second to the font size, another important indicator of font emphasis is the additional styling that was added to the text. Small style changes are often applied to emphasis the importance of single words or sentences. Table XIII outlines the scoring of Bold (Bo), Underline (un) and Italics (It).

TABLE XIII. BOLD ITALICS UNDERLINE

Tool	Bo Yes	Bo No	Bo N.A.	It Yes	It No	It N.A.	Un Yes	Un No	Un N.A.
pdftohtml	35%	51%	14%	20%	45%	35%	4%	53%	43%
unoconv	49%	31%	20%	58%	4%	38%	56%	6%	38%
adobe	86%	4%	10%	62%	6%	32%	45%	14%	41%

Positioning The positioning of the text was varied greatly by the type of tool that was used. Table XIV shows that pdftohtml had the best positioning. This is generally attributed to the fact that it primarily focuses on the layout of the documents.

TABLE XIV. TEXT POSITIONING

Tool	Yes	No
pdftohtml	98%	2%
unoconv	49%	51%
adobe	37%	63%

Sentence Breaking An issue that arose during the preliminary testing was that the sentences and paragraphs were being broken down unnecessarily. Table XV shows that pdftohtml is identified as having the highest level of sentence breaking.

TABLE XV. SENTENCE BREAKING

Tool	Yes	No
pdftohtml	98%	2%
unoconv	2%	98%
adobe	2%	98%

Generation Errors Overall the amount of garbage random character sequences generated on output was quite low. Table XVI shows the high score for the Adobe tool is attributed to its accuracy in attempting to translate text in images to their text representation.

TABLE XVI. GENERATION ERRORS

Tool	Yes	No
pdftohtml	2%	98%
unoconv	2%	98%
adobe	12%	88%

In some cases semantics can be broken by some of the generated errors. pdftohtml scored 2% yes, 98% no. unoconv scored 0% yes, 100% no and finally Adobe scored 4% yes, 96% no.

Correct Document Layout Table XVII identifies pdftohtml as having the best layout overall. The representation by unoconv was generally unreliable, fitting the text to a header and body layout. Adobe had a tendency to group sections of text together, although logically correct was not visually in the correct position.

TABLE XVII. DOCUMENT LAYOUT

Tool	Yes	No
pdftohtml	98%	2%
unoconv	67%	33%
adobe	70%	30%

An issue that arose from the layout was two column layouts being merged together or text from the opposite side of the page merging with other text. The layout breaking semantics with pdftohtml was 4% yes, 96% no. unoconv scored 4% yes, 96% no and Adobe with 14% yes, 86% no.

HTML Structures Three of the main HTML structures that are seen through the documents are bullet lists, numbered lists and tables. Three tests were created to assess the quality of these structures in the outputted file. For all three tools, outputted bullet lists were relative to their collection e.g. correctly in place. Little to no bullets were missing from these lists for all tools. Although outputted as a list, the HTML behind the bullet list was not always the ideal HTML markup, and an alternate representation was used. Table XVIII outlines a test to assess if the outputted HTML was actually a valid HTML.

TABLE XVIII. VALID HTML STRUCTURES

Tool	Bullet List Valid	Number List Valid	Table Valid
pdftohtml	0%	0%	0%
unoconv	92%	39%	53%
adobe	62%	92%	63%

3) Results Review: A noticeable difference can be seen in the quality of data that is extracted from from standard document types e.g. .doc .docx as apposed to .ppt and .pptx files. The complexity and range of arrangements of text and structures that are found in presentation formats varies greatly. In contrast to this, document files have a very standard formatting, that is generally followed throughout the documents.

The complexity of additional images and image based text representations can be seen in presentation formats again, where images are used less frequently in standard document formats.

VI. CONCLUSIONS

During the process of forensic text analysis, the quantity of different file types available creates the need for a similar amount of tailored tools to be developed to extract the text for forensic analysis. This multi-filetype issue can cause particular files to be ignored if filetype specific tools are not available during the extraction and text analysis process.

This paper analyzed a number of document to HTML conversion tools to create HTML formatted document counterparts to allow universal tools to be used during the forensic text analysis process.

To gauge the quantity of loss and quality of text output for each tool, two experiments were conducted. The first gained an overview of possible tools that produced detailed output. The second experiment performed an in depth analysis of the selected tools. The results of the Experiment 1 preliminary tests showed that pure OCR and DLA solutions did not provide the detailed output needed. Once an understanding of the document itself was obtained, better results were produced.

Experiment 2 outlined that although pdftohtml does produce very high quality visual outputs, the overall quality of the HTML that was produced was bulky and not detailed enough to gain any great understanding of the structure. unoconv did produce good output for most files, but the overall detail of the tool was lacking when complex content such as image based representations were presented, resulting in missing output.

Adobe Acrobat Pro by far, produced the most accurate output. Although the layout of the content was not always exact, the quality was generally a lot higher compared to other tools. Acrobat made greatest attempts at converting images to text, which although in some cases produced additional junk output, generally gained more output text. A number of characteristics of the documents, such as two column layout, rotated text and complex image based text routinely caused difficulty, although not entirely impossible to decipher.

REFERENCES

[1] F. Yuan, B. Liu, and G. Yu, "A study on information extraction from pdf files," in *Advances in Machine Learning and Cybernetics*, ser. Lecture Notes in Computer Science, D. Yeung, Z.-Q. Liu, X.-Z. Wang, and H. Yan, Eds. Springer Berlin Heidelberg, 2006, vol. 3930, pp. 258–267.

[2] F. Rahman and H. Alam, "Conversion of pdf documents into html: a case study of document image analysis," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 1, 2003, pp. 87–91 Vol.1.

[3] S. P. et al, "Information extraction tools for portable document format," *International Journal of Computer Technology And Applications*, vol. 2, no. 6, pp. 2047–2051, November-December 2011.

[4] C. Wang, C. Sun, L. Lin, and X. Wang, "A block segmentation based approach for web information extraction," in *Asian Language Processing (IALP), 2010 International Conference on*, dec. 2010, pp. 154 –157.

[5] Y. hui Feng, Y. Hong, W. Tang, J. Yao, and Q. ming Zhu, "Using html tags to improve parallel resources extraction," in *Asian Language Processing (IALP), 2011 International Conference on*, nov. 2011, pp. 255 –259.

[6] J. L. Hong, E.-G. Siew, and S. Egerton, "Viwer- data extraction for search engine results pages using visual cue and dom tree," in *Information Retrieval Knowledge Management, (CAMP), 2010 International Conference on*, march 2010, pp. 167 –172.

[7] J. Zou, D. Le, and G. Thoma, "Combining dom tree and geometric layout analysis for online medical journal article segmentation," in *Digital Libraries, 2006. JCDL '06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, june 2006, pp. 119 –128.

[8] A. Kae and E. Learned-Miller, "Learning on the fly: Font-free approaches to difficult ocr problems," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, july 2009, pp. 571 –575.

[9] T. Breuel, "The hocr embedded ocr workflow and output format," December 2007. [Online]. Available: http://docs.google.com/View?docid=dfxcv4vc_67g844kf