Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers

Christopher Clark and **Santosh Divvala**
The Allen Institute for Artificial Intelligence

http://pdffigures.allenai.org

Abstract

Identifying and extracting figures and tables along with their captions from scholarly articles is important both as a way of providing tools for article summarization, and as part of larger systems that seek to gain deeper, semantic understanding of these articles. While many "off-the-shelf" tools exist that can extract embedded images from these documents, e.g. PDFBox, Poppler, etc., these tools are unable to extract tables, captions, and figures composed of vector graphics. Our proposed approach analyzes the structure of individual pages of a document by detecting chunks of body text, and locates the areas wherein figures or tables could reside by reasoning about the empty regions within that text. This method can extract a wide variety of figures because it does not make strong assumptions about the format of the figures embedded in the document, as long as they can be differentiated from the main article's text. Our algorithm also demonstrates a caption-to-figure matching component that is effective even in cases where individual captions are adjacent to multiple figures. Our contribution also includes methods for leveraging particular consistency and formatting assumptions to identify titles, body text and captions within each article. We introduce a new dataset of 150 computer science papers along with ground truth labels for the locations of the figures, tables and captions within them. Our algorithm achieves 96% precision at 92% recall when tested against this dataset, surpassing previous state of the art. We release our dataset, code, and evaluation scripts on our project website for enabling future research.

1 Introduction

Mining knowledge from documents is a commonly pursued goal, but these efforts have primarily been focused on understanding text. Text mining is, however, an inherently limited approach since figures[1] often contain a crucial part of the information scholarly documents convey. Authors frequently use figures to compare their work to previous work, to convey the quantitative results of their experiments, or to provide visual aids to help readers understand their methods. For example, in the computer science literature it is often the case that authors report their final results in a table or line plot that compares their algorithm's performance against a baseline or previous work. Retrieving this crucial bit of information requires parsing the relevant figure, making purely text based approaches to understanding the content of such documents inevitably incomplete. Additionally, figures are powerful summarization tools. Readers can often get the gist of a paper by glancing through the figures which frequently contain both experimental results and explanatory diagrams of the paper's method. Detecting the associated caption of the figures along with their mentions throughout the rest of the text is an important component of this task. Captions and mentions help provide users with explanations of the graphics found and, for systems that seek to mine semantic knowledge from documents, captions and mentions can help upstream components determine what the extracted figures represent and how they should be interpretted.

Extracting figures requires addressing a few important challenges. First, our system should be ambivalent to the content of the figures in question, which means it should be able to extract figures even if they have heavy textual components, or are entirely composed of text. Therefore our algorithm needs to be highly effective at deciding when text is part of a figure or part of the body text. Second, we need to avoid extracting images that are not relevant (such as logos, mathematical symbols, or lines that are part of the paper's format), and to group individual graphical and textual elements together so they can all be associated as being part of the same figure. Finally, we seek to both identify captions and correctly assign figures and tables to the correct caption. Neither of these tasks is trivial due to the wide variety of ways captions can be formatted and the fact that individual captions can be adjacent to multiple figures making it ambiguous which figure they are referring to.

Our work demonstrates how these challenges can be overcome by taking advantage of prior knowledge of how scholarly documents are laid out. We introduce a number of novel techniques, including i) a method of removing false positives when detecting captions by leveraging a consistency assumption, ii) heuristics that are effective at separating body text and image text, and iii) the insight that a strong source of signal for detecting figures is the location of 'negative space' within the body text of a document. Unlike most previous work in this field, our system is equally effective at both table and figure detection.

Our system (pdffigures) takes as input scholarly documents in PDF form[2]. It outputs, for each figure, the bounding box of that figure's caption and a bounding box around the region of the page that contains all elements that caption refers to. Additionally we expect the correct identifier

[1] Throughout this paper we use the term 'figures' to refer to both tables, figures and their associated captions

[2] We assume the PDF format as it has become the de facto standard for scholarly articles

of the figure to be returned (for example, “Table 1” or “Figure 1”) so that mentions of that figure can be mined from the rest of the text. Previous work on figure extraction has focused on documents from the biomedical (Lopez and others 2011), chemistry (Choudhury and others 2013) or high energy physics (Praczyk and Nogueras-Iso 2013) literature. In this work we study the problem of figure extraction for the domain of computer science papers. Towards this end we release a new dataset, annotated with ground truth bounding boxes, for this domain.

2 Background

At a high level, PDF files can be viewed as a series of operators which draw individual elements of the document. These operators can include text operators, which draw characters at particular coordinates with particular fonts and styles, vector graphic operators that draw lines or shapes, or image operators that draw embedded image files stored internally within the PDF. Within a document, a figure can either be encoded as an embedded image, or a number of embedded images arranged side-by-side, or as a combination of vector graphics and text, or a mix of all three. In more recent computer science documents most plots and flow charts are composed of vector graphics. While many “off-the-shelf” tools exist that can extract embedded images from PDFs, such as PDFBox[3] or Poppler (Poppler 2014), these tools do not extract vector graphic based images. Additionally, such tools leave the problem of caption association unsolved and are liable to extract unwanted images such as logos or mathematical symbols. Image segmentation tools such as the ones found in Tesseract (Smith 2007) or Leptonica[4] are also inadequate. Such tools frequently misclassify regions of the document that contain very text heavy figures as being text, or misclassify bits of text as being images if that text is near graphical elements or contains mathematical symbols.

Figure extraction has been a topic of recent research interest. In (Choudhury and others 2013) captions were extracted using regular expressions followed by classification using font and textual features. We build upon this idea by additionally leveraging a consistency assumption, that all figures and tables within a paper will be labelled with the same style, to prune out false positives. Their system also extracts figures by detecting images embedded in the PDF, however it does not handle vector graphics. Work done by (Lopez and others 2011) identified captions through regular expression followed by clustering the resulting figure mentions. Graphical elements were then clustered by parsing the PDF files and analyzing the individual operators to find figures. While their approach reached 96% accuracy on articles from the biomedical domain their method does not handle figures composed of combinations of image operators and text operators, which is prevalent in the domain of computer science papers. Work by (Praczyk and Nogueras-Iso 2013) similarly detects figures and captions by merging nearby graphical and textual elements while heuristically filtering out irrelevant elements found in the document. Their approach is similar to ours in that an attempt is made to classify text as being body text or as part of a figure to allow the extraction of text heavy figures. However, their algorithm requires making the

[3] http://pdfbox.apache.org/

[4] http://www.leptonica.com/

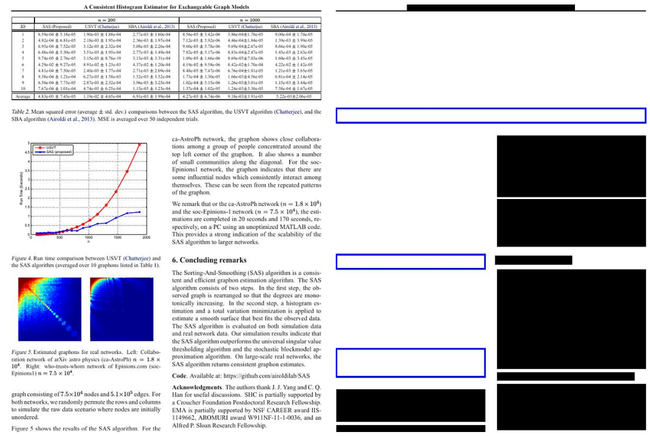

Figure 1: A scholarly document (left, page from (Chan and Airoldi 2014)), and the same document with body text masked with filled boxes, captions masked with empty boxes, and tables and figures removed (right). By examining the right image it is easy to guess which regions of the page each caption refers to, even without knowing the text, what the captions say, and anything about the graphical elements on the page.

assumption that figures, while possibly containing text, contain at least some non-trivial graphical elements to ‘seed’ clusters of elements that compose figures. Our work avoids making this assumption and can thus handle a greater variety of figures, as well as generalize easily to tables. Our work also addresses the problems the can arise when captions are adjacent to multiple figures or multiple figures are adjacent to each other.

Extracting tables has also been addressed as a separate task of great interest in the information extraction community. Our work is not directly comparable since the system presented here locates tables but does not attempt to organize their text into cells in order to fully reconstruct them. Nevertheless locating tables within documents is a non-trivial part of this research problem. Most approaches use carefully built heuristics based on detecting columns of text, vertical and horizontal lines, or white space. A survey can be found at (Zanibbi, Blostein, and Cordy 2004) or in the results of a recent competition (Khusro, Latif, and Ullah 2014). Our work shows that table detection can be completed without relying on hand crafted templates or detailed table detection heuristics by exploiting more general assumptions about the format of the documents being parsed.

A number of search engines for scholarly articles have made attempts to integrate table and figure information. CiteSeerX (Wu and others 2014) extracts and indexes tables in order to allow users to search them, but does not handle figures. The Yale Image Finder (Xu, McCusker, and Krauthammer 2008) allows users to search a large database of figures, but does not automatically extract these figures from documents.

3 Approach

Our algorithm leverages a simple but important observation that if a region of the page does not contain body text and is adjacent to a caption, it is very likely to contain the figure

being referred to by that caption. Figure 1 illustrates how effective this concept can be; much of the time a human reader can locate regions containing figures within a page of a scholarly document even if all that is known is the locations of the captions and body text. This observation stems from our knowledge of how scholarly documents are typically formatted. Authors present information in a continuous flow across the document and so do not include extraneous elements or unneeded whitespace. Therefore regions in the document that do not contain body text must contain something else of importance, almost certainly a figure. A particular motivation for our approach is that figures in scholarly documents are the least structured elements in the document and thus the trickiest to parse effectively. Graphics can have large amounts of text, large amounts of white space, be composed of many separate elements, or otherwise contain content that is hard to anticipate. Tables can be formatted in grids, with only vertical lines, with no lines at all, or in many other variations. However most academic venues have strict guidelines on how body text and section titles should be formatted which tend to follow a narrow set of conventions, such as being left aligned and being in either a one or two column format. Such guidelines make positively identifying body text a much easier task. Once the body text is found, the regions of the document containing figures can be detected without making any assumptions as to the nature of those figures other than that they do not contain any elements that were identified as body text.

Our proposed algorithm has three phases:

1. Caption start identification. This step involves parsing the text of the document to find words like 'Figure 1:' or 'Table 1.' that indicate the start of a caption, while taking steps to avoid false positives. The scope of this phase is limited to identifying the first word of the caption, not the entire caption itself.
2. Region identification. This involves chunking the text in the PDF into blocks, then identifying which blocks of text are captions, body text, or part of a figure. This step also attempts to identify regions containing graphical components. The output is a number of bounding boxes labelled as body text, image text, caption text, or graphic region.
3. Caption assignment. This phase involves assigning, for each caption, the region of space within the document that it refers to, by making use of the regions found in the previous step.

Caption Start Identification

This phase of the algorithm identifies words that mark the beginning of captions within the document. We extract text from documents using Poppler (Poppler 2014). This step assumes that the PDFs being used as input have their body text and captions encoded as PDF text operators, not as part of embedded images. This assumption is almost always true for more recent scholarly PDFs, but exceptions exist for some older PDFs, such as those that were created by scanning paper documents[5].

The extracted text is scanned to find words of the form ⟨Figure| Fig|Table⟩ followed by either a number, or a period or colon and then a number. These phrases are collected as potential starts of captions. This first pass has high recall, but can also generate false positives. To remove false positives, we look for textual cues in combination with a simple consistency assumption: we assume that authors have labelled their figures in a consistent manner as is required by most academic venues. If we detect redundancy in the phrases found, for example if we find multiple phrases referring to 'Figure 1', we attempt to apply a number of filters that selectively remove phrases until we have a unique phrase for each figure mentioned. Filters are only applied if they would leave at least one mention left for each figure number found so far. We have been able to achieve high accuracy using only a handful of filters. These include: (I): Select only phrases that contain a period. (II): Select only phrases that contain a semicolon. (III): Select only phrases that have bold font. (IV): Select only phrases that have italic font. (V): Select only phrases that are of different font sizes than the words that follow them. (VI): Select only phrases that begin new lines, as judged by Poppler's text detection system.

Our filters can be noisy, for example selecting bold phrases can, in some papers, filter out the true captions starts while leaving incorrect ones behind. Detecting bold and italic font can itself be challenging because such fonts can be expressed within a PDF in a variety of ways. However we can usually detect when a filter is noisy by noting that a filter would remove all mentions of a particular figure, in which case the filter is not applied and a different filter can be tried to remove the false positives. In general we have found our caption identification system to be highly accurate, but occasionally our consistency assumption is broken which can lead to errors.

Region Identification

Having detected the caption starts, this phase identifies regions of the document that contain body text, caption text, figure text, or graphical elements. The first step in this phase is identifying blocks of continuous text. To do this we use the text grouping algorithm made available in Poppler (Poppler 2014) to find lines of text within each page. Individual lines are then grouped together by drawing the bounding boxes of each line on a bitmap, expanding these boxes by slight margins, and then running a connected component algorithm[6] to group nearby lines together.

Having identified text blocks we need to decide if those blocks are body text, captions, or part of a figure. We identify caption text by finding text blocks that contain one of the previously identified caption starts and labeling those blocks as captions. We have found it useful to post process these blocks by filtering out text that is above the caption word or not aligned well with the rest of the caption text. The remaining blocks are classified as body text or image text. To identify body text we have found an important cue is the page margins. Scholarly articles align body text down to fractions of an inch to the left page margin, while figure text is often allowed to float free from page margins. We locate margins by parsing the text lines throughout the entire document and detecting places where many lines share the same starting x coordinate. Text blocks that are not aligned with the mar-

[5]Using an OCR system might allow us to put such documents in the same pipeline, albeit with more noise, but is not currently implemented.

[6]http://www.leptonica.com/

Figure 2: Classifying regions within a scholarly document. All text in the document (first panel, page from (Neyshabur and others 2013)) is located and grouped into blocks (second panel). Next the graphical components are isolated and used to determine regions of the page that contain graphics (third panel). To build the final output (fourth panel) these two elements are put together and each text block is classified as body text (filled boxes), image text (box outlines), or caption (box outlines).

gins are labelled as figure text. Left aligned blocks of text that are either small, or tall and narrow, are also classified as figure text because they are usually axis labels or columns within tables that happen to be aligned to the margins. Section headers, titles, and page headers are handled as special cases. To detect pages headers we scan the document to determine whether pages are consistently headed by the same phrase (for example, a running title of the paper might start each page) and if so label those phrases as body text. A similar procedure is used to detect if the pages are numbered and, if so, classify the page numbers found as body text. Section headers are detected by looking for text that is bold, and either column centered or aligned to a margin.

This phase also identifies regions containing graphical elements. To this end, we render the PDF using a customized renderer that ignores all text commands. We then filter out graphics that are contained or nearly contained by a body text region in order to remove lines and symbols that were used as part of a text section. Finally we use the bounding boxes of the connected components of the remaining elements to denote image regions. Figure 2 shows the steps that make up this entire process.

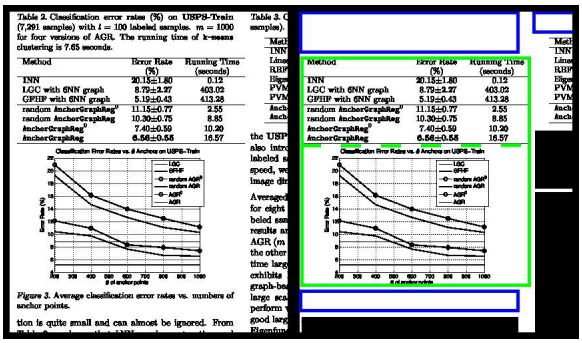

Figure 3: Example of a figure and table being directly adjacent (left panel, from (Liu, He, and Chang 2010)). In this case the proposed figure regions for each caption will by identical and encompass both the plot and table (right, solid lines). We handle this case by detecting that the region is divided in the middle by a section of whitespace, and then splitting the proposed figure region across that whitespace (dashed line).

Figure Assignment

In this final phase we assign each caption to a region of space within the document. Our algorithm is based on the observation that the region of space a figure occupies is almost always both adjacent to one side of its caption and has a box like shape. This algorithm has three parts. Region proposal, which generates, for each caption, potential regions of the document that caption could refer to. Region scoring, which is a function that gives each region a score reflecting how likely it is that the region contains a figure. Finally region selection, which uses the proposed regions and the scoring function to select a proposed region for each caption.

Region proposal is performed by building four regions adjacent to each caption by first expanding that caption's bounding box either to the left, right, up or down as far as possible without running into the body text or the page margin. These regions are then further expanded in the orthogonal directions (for example, boxes would be expanded to the left and right if they were created by expanding the caption's bounding box up or down) as much as possible without running into page margins or body text. We employ one additional heuristic, in two column papers we do not allow the box to cross the center of the page during the second expansion stage unless the caption itself spans both columns. Completing this procedure for each possible expansion direction results in four proposed figure regions for each caption.

Region scoring is a function that gives each region a score

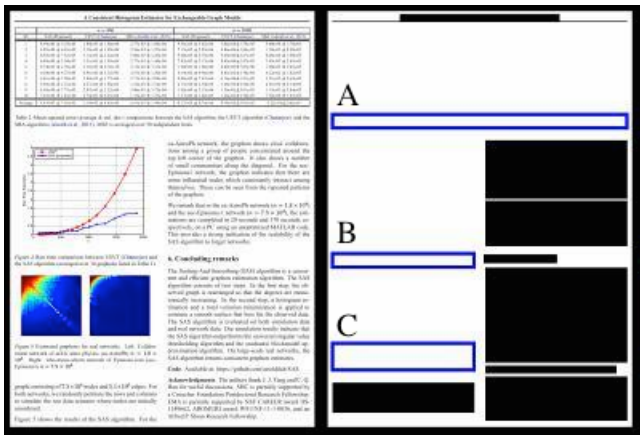

Figure 4: Disambiguating caption to empty space pairing. From the original document (left panel, page from (Aziz and others 2011)) text regions and caption regions are detected (shown as filled and empty boxes in the right panel). At this point it is ambiguous what space to assign to the middle caption, labelled as 'B', because considered in isolation this caption could plausibly refer to the region above or the region below it. However our algorithm detects that the lower caption, caption C, only has one large, empty region of space nearby that it could refer to. Once it is known that that space has to be assigned to caption C it becomes clear caption B must be referring to the region above it.

based on how likely it is to contain a figure. Our scoring function rejects regions that contain only empty pixels, or if they are too small. Otherwise regions are scored based on their size, plus a bonus if they contain a large graphical region and a smaller bonus if they contain a smaller graphical region or many different graphical regions. This heuristic helps us to be robust to errors in the previous steps. For example, if a caption has a line plot above it and a bulleted list below it (that was classified as image text), our scoring function will find it preferable to select the line plot's region over the bulleted list's region as that caption's figure region. However if no image regions are found, for example, if the caption really is about a bulleted list, the scoring function will then allow captions to be assigned to regions that contain nothing but text.

Region selection determines which of the proposed regions to select for each caption. A naive approach would be to iterate through each caption and select its highest scoring region, however this can lead to errors in the face of ambiguities. Consider Figure 4, where captions are adjacent to multiple figures making it impossible to judge which region to assign to each caption when considered in isolation. To deal with such ambiguity we iterate over all possible matchings of accepted figure regions to captions and select the highest scoring configuration. In practice the number of possible configurations is almost alway very small (less than 5) since most pages have only a few figures on them, and each figure typically only has a few proposed figure regions that do not get rejected by the scoring function. Use of this strategy for figure-caption assignment rather than iteratively assigning the highest scoring proposal to its corresponding caption increases our precision by about 2.5% and recall by about 1.5% in our analysis.

	Precision	Recall	F1
Ours	0.957	0.915	0.936
Praczyk and Nogueras-Iso	0.624	0.500	0.555
pdfimages	0.198	0.116	0.146

Table 1: Precision and recall on figure extraction.

	Precision	Recall	F1
Ours	0.952	0.927	0.939
Praczyk and Nogueras-Iso	0.429	0.363	0.393

Table 2: Precision and recall on table extraction.

A troublesome case for our region proposal method is when figures are directly adjacent to each other, in which case it is difficult to tell where one figure ends and where the other one begins. This is shown in Figure 3. In these cases proposed regions might get expanded too much and include multiple figures. To handle this problem, during the region selection stage, if we detect proposed regions would overlap, an attempt is made to split the conflicting region based on areas of whitespace inside the overlap. We found this increases our recall by about 2%.

4 Dataset

We have assembled a new dataset of 150 documents from three popular computer science conferences. We gathered 50 papers from NIPS 2008-2013, 50 from ICML 2009-2014, and 50 from AAAI 2009-2014 by selecting 10 published papers at random from each conference and year. Annotators were asked to mark bounding regions for each figure and caption using LabelMe[7]. For each region marked by annotators, we found the bounding box that contained all foreground pixels within that region. These bounding boxes were then used as ground truth labels. In total we acquired bounding boxes for 458 figures and 190 tables. Our dataset along with the annotations can be downloaded at pdffigures.allenai.org. We hope our new dataset and ground truth annotations will provide an avenue for researchers to develop their algorithms and make comparisons.

5 Results

We assess our proposed algorithm on our dataset and compare its performance to previous methods. We expect the system being evaluated to return, for each figure extracted, a bounding box for both the figure and it's caption as well as the identifier of that figure (e.g., "Figure 1" or "Table 3") and the page number that the figure resides on. Figures with identifiers that did not exist in the hand built labels, or with incorrect page numbers, are considered incorrect. Otherwise a figure is judged by comparing the bounding boxes returned against the ground truth using the overlap score criterion from (Everingham and others 2010). Boxes are scored based on the area of intersection of the ground truth bounding box and the output bounding box divided by the area of the union between them. If the overlap score exceeds 0.80, we consider the output box to be correct, otherwise it is marked as incorrect. Caption box regions are scored using the same

[7] http://labelme2.csail.mit.edu/Release3.0/index.php

criterion. We consider an extraction to be correct if both the caption and the figure bounding boxes are correct according to the above criterion. Following this setup we evaluate our system and compare to the work of (Praczyk and Nogueras-Iso 2013). The work by (Praczyk and Nogueras-Iso 2013) does not return figure identifiers so we used regular expression to extract identifiers based on the first word of the caption text.

We also evaluate pdfimages (Poppler 2014), a popular tool for extracting embedded images from PDFs. We ran this tool on our dataset and filtered out the extracted images that were smaller than a square inch. We evaluate the results in this case using a much more lenient scoring criterion. We mark an extraction as correct if its size is within 80% of an annotated figure region on the corresponding page and wrong otherwise. Results of our evaluation on figures and tables can be found in Table 1 and Table 2 respectively. The outputs and evaluation of both our algorithm and the system by (Praczyk and Nogueras-Iso 2013) are available on our project website.

Our algorithm obtained around 96% precision at 92% recall for tables and figures. Despite being leniently scored, pdfimages performed extremely poorly. It is capable of getting correct results if a figure is encoded as a single embedded image within a document, but this is so rare in computer science papers, it was unable to even get 15% recall. The algorithm by (Praczyk and Nogueras-Iso 2013), although achieving high results on the domain of high energy physics (HEP), did not generalize well to the domain of computer science papers, achieving much lower recall and precision. Errors from (Praczyk and Nogueras-Iso 2013) are often due to mishandling cases where figures are close together, or allowing body text to be grouped into figure regions. In particular for papers from ICML, figure regions often included body text from the opposite column. Mistakes detecting captions were also a significant source of error. This indicates some of the heuristics used by (Praczyk and Nogueras-Iso 2013) failed to generalize from the HEP domain to computer science papers.

We also analyzed the sources of errors of our approach. Approximately a half of the errors produced by our algorithm were caused by non-standard formatting, such as, non-standard caption titles (e.g., Using "Figures 1 and 2" instead of "Figure 1:" and "Figure 2:"), or using small fonts for captions, or PDFs containing large numbers of extraneous operators that were detected by the text extraction tool (Poppler 2014) but were not visible on the document. The remaining half were caused by various text blocking errors, misclassifying blocks of text, or being unable to split regions containing multiple figures correctly.

Our algorithm takes less than two seconds to process a paper[8], and is therefore scalable to large datasets. In fact, we have run our method on 21,000 documents from the ACL corpus (Bird and others 2008) . The results are available on our project website.

6 Discussion

In this paper, we presented a novel approach (pdffigures) for extracting figures and tables from computer science papers. The contributions of our work include a method of identifying captions by exploiting the fact that captions are consistently formatted in scholarly documents, heuristics that are effective at separating body text and image text, and the insight that we can identify figures using the 'negative space' within body text as well the graphical elements within the PDF. We additionally released a novel dataset of computer science papers for figure extraction along with their ground truth labels. For future work, the figure assignment algorithm could be improved by using a more sophisticated scoring method or considering a wider diversity of region proposals for each caption, which could be used to provide resilience to errors in the previous steps. Finally, more carefully accounting for graphical elements, possibly integrating the kinds of clustering techniques that were used by (Praczyk and Nogueras-Iso 2013), might provide an avenue to improve results.

[8]On a single thread on a Macintosh OS X 10.9 with a 2.5GHz Intel core i7.

7 Acknowledgments

We thank Isaac Cowhey for annotating our dataset. We would also like to thank Oren Etzioni, Peter Clark, Isaac Cowhey, and Sam Skjonsberg for their helpful comments and reviews.

References

Aziz, et al. 2011. False-name manipulations in weighted voting games. *Journal of Artificial Intelligence Research*.

Bird, et al. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*.

Chan, S. H., and Airoldi, E. M. 2014. A consistent histogram estimator for exchangeable graph models. *arXiv preprint arXiv:1402.1888*.

Choudhury, et al. 2013. Figure metadata extraction from digital documents. In *ICDAR*.

Everingham, et al. 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*.

Khusro, S.; Latif, A.; and Ullah, I. 2014. On methods and tools of table detection, extraction and annotation in pdf documents. *Journal of Information Science*.

Liu, W.; He, J.; and Chang, S.-F. 2010. Large graph construction for scalable semi-supervised learning. In *ICML*.

Lopez, et al. 2011. An automatic system for extracting figures and captions in biomedical pdf documents. In *BIBM*.

Neyshabur, et al. 2013. The power of asymmetry in binary hashing. In *NIPS*.

Poppler. 2014. Poppler. http://poppler.freedesktop.org/. Accessed: 2014-09-24.

Praczyk, P. A., and Nogueras-Iso, J. 2013. Automatic extraction of figures from scientific publications in high-energy physics. *Information Technology and Libraries*.

Smith, R. 2007. An overview of the tesseract ocr engine. In *ICDAR*.

Wu, et al. 2014. Citeseerx: AI in a digital library search engine.

Xu, S.; McCusker, J.; and Krauthammer, M. 2008. Yale image finder (YIF): a new search engine for retrieving biomedical images. In *Bioinformatics*. Oxford Univ Press.

Zanibbi, R.; Blostein, D.; and Cordy, J. R. 2004. A survey of table recognition. In *ICDAR*.