2013 12th International Conference on Document Analysis and Recognition

Figure Metadata Extraction From Digital Documents

Sagnik Ray Choudhury[†], Prasenjit Mitra[†‡], Andi Kirk[⋆], Silvia Szep[⋆], Donald Pellegrino[⋆], Sue Jones[⋆], C. Lee. Giles[†‡]

[†]Information Sciences and Technology, [‡]Computer Science and Engineering,
The Pennsylvania State University, University Park, PA 16802 USA
[⋆]The Dow Chemical Company, Spring House, PA 19477 USA
sagnik@psu.edu, pmitra@ist.psu.edu, {andikirk,sszep,dapellegrino,susanjones}@dow.com, giles@ist.psu.edu

***Abstract*—Academic papers contain multiple figures (information graphics) representing important findings and experimental results. Automatic data extraction from such figures and classification of information graphics is not straightforward and a well studied problem in document analysis[6]. Also, very few digital library search engines index figures and/or associated metadata (figure caption) from PDF documents. We describe the very first step in indexing, classification and data extraction from figures in PDF documents - accurate automatic extraction of figures and associated metadata, a nontrivial task. Document layout, font information, lexical and linguistic features for figure caption extraction from PDF documents is considered for both rule based and machine learning based approaches. We also describe a digital library search engine that indexes figure captions and mentions from 150K documents, extracted by our custom built extractor.**

I. Introduction

Figures in documents are rich sources of information and there has long been interest in the problems of classification and automatic extraction of data from such figures. Many such documents are in PDF formats. Although figures are of such importance, except for a few (such as Yale Image Finder[1], BioText[2] and askHermes[3]), most digital libraries do not allow users to search specifically for figures in their documents. Currently, available figure search engines index figures in documents from the PMC[4] repository, which provides a dataset of images and associated metadata for figures appearing in documents. But in most cases, academic document repositories have only the PDF file for a document, from which figures and associated metadata will have to be extracted.

We describe an extraction process for extracting figures and their associated metadata (caption, mentions) from PDF documents. We utilize document layout and font information based features along with lexical features to identify figure captions inside a document. We also design and develop a scalable Solr/Lucene based figure metadata search engine built on top of extracted figure metadata from chemistry journal papers. However, these methods will also work for other scientific domains. Our search system uses a modified ranking function of Lucene to improve the quality of search results.

II. Related Work

Classification of figures in academic documents has been explored extensively[6], [8]. Figures were analyzed extensively, with attempts to vectorize raster images[2] or extract data from 2D plots and solid line curves[5]. However, these work did not address the figure and metadata extraction process itself. For example, [5] extracted text below the figure to use as textual features for classification, but accurate extraction of figure caption was not investigated.

Recently a method [4] was proposed for extraction of images and captions from PDF files with images extracted from PDF documents using Xpdf[5]. Captions are extracted using regular expressions and filters. A figure caption would be a paragraph starting with the term “Fig.” or “Figure”. Extracting paragraphs from PDF documents by parsing the document is mentioned but not explained in detail. Since the PDF document manual[6] does not explicitly mention operators for identifying paragraph boundaries, we believe that they used structural information such as coordinates of text. Apart from structural information, we explore several features for paragraph segmentation. Since all paragraphs starting with the term “Fig.” or “Figure” are not actual captions, extraneous paragraphs need to be removed, say by a filter [4]. Extracted captions are matched with images using structural and geometric cues.

Bhatia and Mitra reported a regular expression based method for extraction of document element captions, which is the first step in our approach[1] . Therefore, our precision should be equal or better than theirs. Search engines on specific document elements such as table [3] or acknowledged entities[7] have been reported earlier. Our system is a continuation of this line of work and more importantly can be readily integrated with other search features; a preliminary description of the search engine and a rule based extractor[7] was recently reported.

III. Extraction Process

A. General Strategy

The process of extraction of figures and associated caption from a document has three sub tasks: 1. Extraction of the image

[1] http://krauthammerlab.med.yale.edu/imagefinder/

[2] http://biosearch.berkeley.edu/

[3] http://figuresearch.askhermes.org

[4] http://www.ncbi.nlm.nih.gov/pmc/

[5] http://www.foolabs.com/xpdf/

[6] http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/PDF32000_2008.pdf

[7] http://ackseer.ist.psu.edu

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.34

file corresponding to the figure, 2. Extraction of caption and 3. Associating figures with captions. Our system is implemented to work on PDF documents.

1) Extraction of figures and text: We use a popular Java based PDF processing library PDFBox to extract text (text lines are extracted sequentially, as they appear in the original file) and raster graphics (image file for the graphic element, location, length and width) from PDF files. PDFBox or other common PDF processing libraries (Xpdf, PyPDF[8]) are not suitable for extraction of vector graphics. Also, they do not extract text from scanned articles, which need to be processed by OCR.

2) Extraction of captions: To extract the captions, we classify a line in the text as a general line, caption beginning line or caption ending line. All lines between caption beginning and ending lines are part of the caption. The caption extraction process is explained later in greater details (section III-B and III-C).

3) Matching captions with figures: We extract text from a rectangle R below the figure to capture the figure id (the number by which the figure is referred in document, such as "Fig 1", "Fig. 2.3", "Figure 5(a)" etc.). Parameters for the rectangle R are:
R_{xy}: x and y co-ordinate of upper left corner.
R_w and R_l: width and length of the rectangle, respectively.

In single-column and multi-column documents where the figure spans multiple columns, R_{xy} is the leftmost point in the page below the figure and the R_w is the page width. In multicolumn documents where the figure spans a single column, R_{xy} is the leftmost point in the column below the figure and R_w is the column width. Length of the rectangle (R_l) is kept at 200 pixels. Part of the caption might be captured inside the box, with some noise. Therefore, we use this text only to identify the figure id and not the whole caption. With a figure with id f_i, we associate the caption starting with "Fig. f_i" or "Figure f_i".

B. A Rule Based Figure Caption Extractor Using Document Layout and Font Based Features

Using document layout and font based features, we developed a rule based extractor system which is reported here. This system extracts the figure and the figure identifier (id) together, and then search for the figure caption starting with the id. The process is summarized in algorithm 1. Steps of the algorithm are explained in greater details here.

1) Preprocessing: : All lines those contain the term "figure" or its lexicographic variations are extracted first and a list L_{rv} is created. While searching for figure captions, L_{rv} is used instead of whole text to increase computational efficiency. For each line, length of the line, font size, and font weight is extracted as features to be used later. Image files corresponding to graphic elements (figures), location and dimensions are also extracted.

```
Data: A PDF document.
Result: Figures, matched with captions and mentions.
From a document d, extract all text lines that contain
the term "figure" or "fig" in a list $L_{rv}$;
for Each line in the whole text of the document do
    Store length of the line, font size, font weight in a
    list) ;
end
for Each figure $f_i$ in the document $d$ do
    Extract figure $f_i$ using PDFBox ;
    Extract the text in a rectangle below the figure $f_i$ ;
    From the extracted text, find out the id $fid_i$ ;
    if no id is extracted then
        Output the image file of the figure ;
        break;
    end
    else
        Caption=Extract_Caption(id);
        Mention=Extract_Mention(id);
        if Caption is null then
            caption=Mention[1]
        end
        Output image file for the figure, metadata file
        for the caption and mention ;
    end
end
```

Algorithm 1: Algorithm for extracting figures, associated caption and mentions from a PDF document.

2) Determination of figure id: : To identify the figure id, following grammar is used over text extracted from a rectangle below the figure:

```
<CAPTION>:= <FIGTERM> <ID>
<ID>:=<NUMID> <CHARID>
<NUMID>:=<DIGIT> (.| <DIGIT>)*
<CHARID>:=(<PUNCT>)* <CHAR> (<PUNCT>)*
<FIGTERM>:= FIGURE|FIG.|Figure|Fig.
<DIGIT>:=0:9
<PUNCT>:=(|[|)|]
<CHAR>:=a:z|A:Z
```

The key assumption here is the caption of the figure should follow the figure. Though this is not always the case, counter examples are not common, especially in scientific papers. Alternatively, figure and caption could be extracted separately and then matched based on distance, which will sacrifice precision of the process.

3) Caption extraction: : A figure caption beginning line should start with the term "Fig." or "Figure". As all such lines will not be an actual caption beginning line such lines need to be filtered. Also, the lines below the caption beginning line need to be filtered to identify the caption ending line. Our filters are based on features extracted from layout and font information.

Line_length(i,j): Returns **true** if length (difference between x coordinate of first and last character) of line i > length of line j by a threshold (10% of the median of line lengths), else returns **false**.

[8] http://knowah.github.com/PyPDF2/

Bold_font(i): Returns **true** if a character in the line i is written in bold font, else returns **false**.

FontSize_change(i,j): Returns **true** if average font sizes of line i and j are different by a threshold (10% of median of average font sizes), else returns **false**.

Usually, caption beginning lines have smaller length than previous line in single column documents and caption ending lines have smaller length than the next line in most figures. Also, most caption beginning lines begin with a bold character (such as **Fig. 1**) and in general, caption lines have a different font size than other lines in text.

Caption beginning line extraction: We first search the list L_{rv} for lines starting with "Figure" or "Fig." followed by an id **i** extracted in the previous step. All matched lines are stored in a list of potential caption beginning lines PC_i. Let CB_i denote the actual caption beginning line for f_i. Following procedure is used to extract CB_i:

```
For each line $l_j$ in $PC_i$
If ((Line_length($l_j$,$l_j$-1) & Bold_font($l_j$)) OR
   (Line_length($l_j$,$l_j$-1) & FontSize_change($l_j$,$l_j$-1)) OR
   (FontSize_change($l_j$,$l_j$-1) & Bold_font($l_j$)))
   $CB_i$=$l_j$;
   break;
```

The first line l_j that passes any two of aforementioned filters is considered a match for CB_i.

Caption ending line extraction: If CB_i is null in the previous step (no line passed through the filters), the caption extraction method returns a null caption. Otherwise, each line, including and after CB_i is checked for length difference or font size difference. The first line that satisfies any one criterion is considered a match for selection as the caption ending line. For figures embedded in the same column, caption lines are extracted sequentially. Therefore, for each line, it is checked whether the next line is a caption beginning line, which would imply that the current line is ending line of another caption.

4) Identifying mentions: : Other lines which contain the term "figure" (or its lexicographic variations) followed by the id f_i are extracted using regular expressions. Seven lines below and above each such line is combined to create the mention metadata for the figure f_i.

C. Lexical Features for Caption Line Extraction

Our rule based system was developed as a scalable method to extract caption lines from non scanned PDF documents with reasonable accuracy. This system uses document layout and font information based features. Extraction of these features is dependent on the underlying library and can vary over different document formats. In this section, we explore linguistic and lexical features for caption beginning and ending line identification, which are more generic and domain independent.

1) Features for caption beginning line identification: Given a text line starting with "Fig." or "Figure", we classify it as a caption beginning line or not. In caption beginning lines, after the id, a new sentence is started, therefore the id is usually followed by a noun phrase. On the contrary, other lines starting with the term "Fig" is a sentence where the noun phrase is the term "Figure id" itself. Therefore, the word after the id is usually a verb. We use an open source part of speech (POS) tagger[9] to tag the line and collect the POS tag of the first word followed by id using the previously mentioned grammar. As POS taggers work on sentence level and we tag part of a sentence, accuracy of the tagging is not high. Therefore, we also consider whether the word followed by the id starts with a capital letter, which is true for most caption beginning lines and false for the opposite case. We also consider the punctuation mark after the numeric id as a feature. Certain punctuations such as ":" after the id almost surely indicates a caption beginning line, whereas punctuations such as ")" are indicative of the other class. The features are summarized below:

POSTag: A binary feature indicating whether the POS tag of the word following the id is a verb or not.

Capitalization: A binary feature indicating whether the word following the id starts with a capital letter or not.

Punctuation: A probability value (1 if the punctuation is ":", 0 if the punctuation is ")", else 0.5) is assigned depending on the punctuation after the numeric id.

2) Features for caption ending line identification: Given lines below the caption beginning line, our goal is to classify them as a caption ending line or not. This binary classification problem is similar to paragraph segmentation problem, where the goal is to identify paragraph boundaries from free text. For paragraph segmentation, several parse tree based and cohesion based features have been explored before, of which two features have been found to be most useful[9]: 1)*Relative position*: length of current sentence from the previous boundary and 2) $Word_1$, $Word_2$: first and second words of the sentence. Other important features include unclosed punctuation, lexical similarity between sentences (lexical cohesion), anaphoric cues (terms such as "this"), nodes of parse tree. Extraction of these feature requires sentence segmentation of text, which itself is a hard problem. Our features are motivated from these features, but different, as we use lines as unit of input instead of sentences. We used following features:

Distance from first line: Probability(a line is a caption ending line) increases as distance from the caption beginning line increases. This feature is assigned a normalized value of 0.5+0.5x(distance from nearest caption beginning line/10) which indicates that a caption beginning line is moderately probable of being a caption ending line itself, whereas, a line with distance 10 is highly probable.

Unclosed punctuation: A caption ending line is obviously a part of the last sentence of a paragraph. Presence of an unclosed punctuation (an open brace, quotation mark) in a line indicates with highly probability that the line is not an caption ending line, but, a line containing no such punctuation is equally probable to be a caption ending line or not. This feature is assigned a normalized value of 1 or 0.5 depending on the previous condition.

Next line first word: Presence of words such as "Figure" or "Table" in the beginning of the next line indicates beginning of a new caption, indicating current line is ending line of

[9] http://nlp.stanford.edu/software/tagger.shtml

another caption whereas other words indicate nothing in particular. Based on this condition, a high probability or moderate probability value is assigned in this feature.

Last character of previous line: Presence of punctuation marks such as ‘,’, ‘;’ or ‘-’ indicates the current sentence has not ended, making it improbable for the current line to be ending line of the caption. Other characters are non discriminative, hence assigned moderate probability.

Character ratio: In a well formatted text, the lines usually have similar number of characters, with the exception of paragraph ending line which has less number of characters than the previous line and next line. This idea is encoded in this feature by assigning a value of $(|c_{l+1}/c_l - c_l/c_{l-1}|)$ where c_l is the number of characters in line l. A larger value indicates a higher probability of being a caption ending line. This “sparse line” feature has been effectively used before in table extraction[3].

The features mentioned here are useful to determine paragraph boundaries in general. As in our system, ten lines below the caption beginning line is classified as caption ending line or not (captions are usually not longer than ten lines), we only expect a single paragraph break and not multiple ones.

IV. Experiments and Results

A. Rule Based System

1) Dataset: To evaluate the rule based system, we randomly selected a set of 300 documents from a set of 150K documents of chemical journal papers. Our rule based extractor was designed to extract <figure, caption> pairs (we did not extract captions if the figure could not be extracted), we manually selected 150 PDF files that did not appear to be scanned articles. Also, these articles primarily contained raster graphics, since PDF processing libraries are not reliable for extraction of vector graphics. Our dataset had articles from eleven chemical journals, which were published between 1997 and 2006.

2) Experiment design: We compared our extractor with another recent one[10] mentioned before[4]. We manually examined the articles and found 883 figures in them had captions and figure identifiers. The evaluation parameters for the experiment follow:

Figure-caption recall: { Number of (figure, caption) pairs extracted } / { Total number of (figure, caption) pairs present in the dataset}.

Caption precision: Ratio of retrieved captions that were correct. We considered a caption to be wrong if 1. There was a mismatch between extracted caption and figure (for example, caption for figure *i* was matched with figure *j*) and 2. Instead of a caption, a paragraph containing a mention to the figure was extracted.

Hard accuracy test: A caption was considered to pass a hard accuracy test if textual similarity between extracted caption and the original caption was $\geq 95\%$. Most captions contained special characters such as α, β or super/subscript characters, which were not extracted properly by the processing library, and, hence, were not considered as matching criteria.

TABLE I. Comparative results of extraction efficiency of our system (E_{pdbx}) with another [4] (E_{Xpdf}). Row 1 reports fraction of (figure, caption) pairs that were retrieved from the dataset. Row 2 reports fraction of retrieved captions that were found correct. Row 3 reports fraction of correct captions that were extracted with $\geq 95\%$ accuracy.

Comparison parameter	E_{pdbx}	E_{Xpdf}[4]
Figure-caption recall	0.84	0.82
Caption precision	0.95	0.90
CE hard accuracy	0.91	0.92

3) Results: Comparative results between our PDFBox based extractor (E_{pdbx}) and Xpdf based extractor (E_{Xpdf}) reported in [4] are summarized in table I.

E_{pdbx} has a recall of 84%, slightly better than E_{Xpdf}. This result is dependent on the underlying software(PDFBox and Xpdf) and does not entirely reflect the relative performance of the caption identification and matching algorithms. However, the results for precision and hard accuracy test do reflect the relative ability of two systems to identify caption lines properly and match them with figures. Our system had better precision on our dataset ([4] reported 95% precision in their experiments), and both extractors extracted captions with comparable accuracy.

B. Machine Learning Based System

1) Dataset: To evaluate the effect of designed features, we randomly selected 150 documents (50 each) from three different data sources: 1) 150K documents in the chemical domain, 2) More than a million documents from Citeseer^X repository, which primarily contains articles from computer science and 3) 2000 documents from arXiv repository, which contains articles in physics and astronomy. The documents were manually examined to ensure that text could extracted. This variation was enforced to test whether the features were writing style and document layout independent.

2) Experiment design: Many documents in our dataset (especially the computer science and physics articles) contained vector graphics, which were not extracted by PDFBox or Xpdf. As the implementation[11] of [4] do not output captions when figures are not extracted (similar to our rule based extractor), we were unable to directly compare the performance of both systems. Instead, we report classification accuracy for caption beginning and ending line identification problems.

From the documents, 3400 lines starting with terms “Fig.” or “Figure” were extracted and manually tagged as caption beginning line or not. 2805 (82.5%) lines were actual caption beginning lines. For investigating the caption ending line problem, 150 actual captions were randomly selected. From ten lines those followed the beginning line in each caption, 850 non caption ending lines were randomly selected and 150 caption ending lines were manually selected.

We trained and tested the model using Matlab[12] based implementation of Naive Bayes and Support Vector Machine (SVM) classifiers (linear and quadratic kernel). SVMs have

[10] http://annotation.dbi.udel.edu/image_mining/index.php

[11] http://annotation.dbi.udel.edu/image_mining/index.php

[12] http://www.mathworks.com/products/matlab/

TABLE II. Accuracy, sensitivity and specificity results are shown for two binary classification problems. Column 1 results correspond to the problem of caption beginning line identification and column 2 results correspond to the problem of caption ending line identification

Comparison parameter	**CBL identification**	**CEL identification**
Accuracy	0.90	0.87
Sensitivity	0.90	0.81
Specificity	0.88	0.87

been extensively used in binary classification problem as it effectively finds the decision surface maximizing margin between data points in two classes. Naive Bayes classifiers have been used before in binary classification tasks related to text categorization. The evaluation parameters follow:
Accuracy: Percentage of caption beginning(ending) lines that were correctly classified.
Sensitivity: { Number of lines classified as caption beginning(ending) line / Number of actual caption beginning(ending) line }.
Specificity: { Number of lines classified as NOT caption beginning(ending) line / Number of lines actually not caption beginning(ending) line. }
Accuracy results were comparable for the SVM and Naive Bayes classifiers. The linear SVM classifier outperformed the Naive Bayes classifier in the sensitivity parameter, specifically in the caption ending line classification problem. Using a quadratic kernel instead of the linear kernel in SVM did not provide much improvements in caption ending line classification, but an average improvement of around 3% was observed in caption beginning line identification problem in all three evaluation parameters. Results for 10 fold cross validation using a linear kernel svm is presented in table II.

3) Results: The results in table II show that document layout and writing style independent lexical features are sufficiently capable of caption line identification. In another experiment, we included a font information based feature (*FontSize_change(i,j)*) used in the rule based system and observed 2 percent improvement on average in both problems. As the "*Character ratio*" feature used here is equivalent of the line length based feature used in the rule based system, it was not repeated. We observed that the font weight information was not extracted properly from many documents in our dataset, therefore refrained from using the *Bold_font(i)* feature in the classification experiment.

V. Search Engine

We ran the rule based system on 150K documents to extract 90,000 figures and associated metadata files, proving its scalability. On top of these metadata files, we built a search engine which allows users to search on caption and mention of the figures present in articles. We used Django[13] to build the user interface. The generic ranking algorithm of Lucene was modified to boost caption fields over mention fields. An important feature of our search engine is "loosely coupled" design, which allows us to change any module independent of others.

VI. Conclusion and Future Work

We proposed an automated process for accurate extraction of figures and associated metadata, a problem that seems to have been largely ignored for PDF documents. We show that accurate figure metadata (specifically, caption) extraction is nontrivial and can be mapped to the paragraph segmentation problem. We explore document layout, font information, lexical, and linguistic features for the problem of caption extraction. By experimenting on document collections from several domains, we show that certain features that are independent of document layout in scholarly domain still have high accuracy. Our figure extraction software is used in extraction of figures and metadata from a large collection of documents, which are then indexed in a scalable Solr/Lucene based digital library search engine which allows users to search on text in figure captions and their mentions in documents. Future work would be to investigate large scale classification and automatic extraction of data from figures in datasets.

VII. Acknowledgements

We gratefully acknowledge partial support from the National Science Foundation and suggestions by Suppawong Tuarob, Shibamouli Lahiri, Lior Rokach, Madian Khabsa and Hung-Hsuan Chen.

References

[1] S. Bhatia and P. Mitra. Summarizing figures, tables, and algorithms in scientific publications to augment search results. *ACM Transactions on Information Systems (TOIS)*, 30(1):3, 2012.

[2] R. Liu, W. Huang, and C. L. Tan. Extraction of vectorized graphical information from scientific chart images. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 521–525. IEEE, 2007.

[3] Y. Liu, K. Bai, P. Mitra, and C. L. Giles. Improving the table boundary detection in pdfs by fixing the sequence error of the sparse lines. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1006–1010. IEEE, 2009.

[4] L. Lopez, J. Yu, C. Arighi, H. Huang, H. Shatkay, and C. Wu. An automatic system for extracting figures and captions in biomedical pdf documents. In *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*, pages 578–581. IEEE, 2011.

[5] X. Lu, S. Kataria, W. J. Brouwer, J. Z. Wang, P. Mitra, and C. L. Giles. Automated analysis of images in documents for intelligent document search. *IJDAR*, 12(2):65–81, 2009.

[6] V. Prasad, B. Siddiquie, J. Golbeck, and L. Davis. Classifying computer generated charts. In *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on*, pages 85 –92, june 2007.

[7] S. Ray Choudhury, S. Tuarob, P. Mitra, L. Rokach, A. Kirk, S. Szep, D. Pellegrino, S. Jones, and C. L. Giles. A figure search engine architecture for a chemistry digital library. *To appear*. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2013.

[8] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 393–402. ACM, 2011.

[9] C. Sporleder and M. Lapata. Automatic paragraph identification: A study across languages and domains. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 72–79, 2004.

[13] https://www.djangoproject.com/