OCR++: An Open-Source Framework For Extracting Information From Scholarly Articles

Samuel Bushi ,Sidhartha Satapathy , Kumar Ayush, Barnopriyo Barua,
Krishna Sai Rohith, Manvi Garg, Priyank Palod, Tulasi Gamidi
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, WB, India
{samuelsbushi,sidharthasatapathy4,k.ayush147,barno0695}@gmail.com
{krishnasai.rohith07,manvi.garg171,priyankpalod,tulasi5091}@gmail.com

ABSTRACT

Scholarly articles are a rich source of scientific research information. Most of the publication venues provide collections of research publications as high quality textual documents in raw PDF format. Information extraction from PDF files is considered as a challenging task mainly due to OCR errors. Current state-of-the-art softwares try to tackle this extraction problem using several commercial tools and complex machine learning models. We introduce *OCR++* , an open-source information extraction framework for scientific publications. We analyze a rich collection of articles in PDF format from different venues. Based on these observations, we propose several information extraction tasks, for example, title, author names, author affiliations, urls, footnotes, sections, headings, references extraction etc. We propose conditional random field based models and hand-written rules for each extraction task. Furthermore, we also map each citation instance with the reference instance. Comparison [1]with the state-of-the-art softwares/tools shows significant improvement in each of the tasks along with fast implementation speed-ups and batch processing functionality.

Keywords

Information Extraction; PDF; XML; Mapping

1. INTRODUCTION

Scientific papers offer a wealth of information that allows researchers to understand research activities in their area of interest. Obtaining structured data from documents is necessary to support retrieval tasks [1]. Significant work has been done in the past to automatically extract information from PDF documents [3, 4, 6]. GROBID [5] performs reliable bibliographic data extractions from scholarly articles combined with multilevel term extractions. Another work, ParsCit [2], is a freely available, open-source implementation of a reference string parsing package which has in its core, a trained conditional random field (CRF) model used to label the token sequences in the reference string.

However, the extraction task is erroneous due to lack of standard formats. Different style guides imposed by various publishers and venues as well as the scope of information provided by individual authors increase the difficulty of automatic information extraction.

In this paper, we describe *OCR++*, an extraction framework to extract textual information as metadata, sections, subsections, headings, table area, figure area, bibliographic references and entities from scholarly articles. The Framework employs a variety of CRF models and hand-written heuristics specialized in different sub-structures. We evaluate *OCR++* with different formatting styles for each of the extraction task.

Although more experimental, we plan to extend the framework with search engine implementation for scientific literature. The search facility will provide richer and better text content and structures than basic PDF extractors.

Next, we describe different extraction tasks and methodologies for each task.

[1]Work in Progress

2. EXTRACTION TASKS AND EVALUATION

We use *pdf2xml*[2] tool to convert PDF files into rich XML files. Each word is annotated with rich metadata, namely, x and y co-ordinates, font size, font weight, font style etc. We perform extraction task using this metadata information and other writing standards. We divide the information extraction tasks into sub-tasks. For each sub-task, we use CRF based model as well as hand-written rules. We evaluate our framework on various scientific writing formats (20 randomly selected PDF documents).Using state of the art machine learning techniques and hand-written heuristics, OCR++ performs reliable data extractions from scholarly articles. We present precision, recall and f-scores for each sub-task as shown in the tables. Next, we describe in detail various sub-tasks with examples:

2.1 Title and Author Name

For title and author names extraction, we define several features. It includes *Boldness, Relative position in the research paper, Relative size of the token, First letter in uppercase, Boldness and size combined, Case of first letter of*

[2]http://sourceforge.net/projects/pdf2xml/

previous token, Case of first letter of current token and first letter of next token.

Table 1:Accuracy results for metadata extraction task

Data	Precision(%)	Recall(%)	F-Score(%)
Email	100	96.1	98.01
Affiliation	95	100	97.43
Title	99.97	94.47	97.14
Author Name	100	79.38	88.56
First Name	99.96	76.47	86.65
Middle Name	99.99	20.00	33.33
Last Name	99.96	76.47	86.65
Author-email mapping	100	69.04	81.69

The reasons for selecting these features are:

- **Title**
  - *Boldness*: Usually, titles are bold.
  - *Relative position*: Mostly, title comes in the beginning of the paper.
  - *Relative size*: Titles generally have the largest font-size in the entire document.
  - *Boldness and Size combined*: To differentiate it from other sub-headings.
  - *First letter in uppercase*: Most of the words in the title start with caps.
- **Author name(First name, Middle name, Last name)**
  - *Relative position*: Names generally come in the beginning of the paper.
  - *First letter in uppercase*: Names always start with caps.
  - *First letter of current token in uppercase and First letter of previous token in uppercase*: Middle name always comes after First name and Last name always comes after Middle/First name.
  - *First letter of current token in uppercase and First letter of next token in uppercase*: Middle name always comes before Last name and First name always comes before Middle/Last name.

2.2 Email and Affiliation

We use hand-written rules to extract email and affiliations. Following are the various formats of writing emails usually found in scholarly articles:

- example1@cse.iitkgp.ernet.in
- {example1, example2, example3 }@cse.iitkgp.ernet.com
- [example4,example5,example6]@cse.iitkgp.ernet.com
- [example7@cse, example7@ee].iitkgp.ernet.in

An Email always has the following properties:

- Author emails are always found on the first two pages.
- An email always has an "@" in it.
- After the "@", there must be period sign (".").
- The period is immediately followed by a lowercase letter(This helps the tool to to distinguish it from a full stop).

Affiliation description generally follows the following formats:

- An affiliation is always more than a single token and contains at least one of the following words starting with capital letter - "University", "Research", "Laboratories", "Corporation", "College", "Institute".
- Sometimes, the university names are in other languages like French. But usually, the word is almost the same. For e.g. University becomes Universite and Institute becomes Institut.
- But the word Research is often found in the Title of research papers. Hence, we first find whether the line has the largest font size in the paper or not (normally of the Title). If it has, then we do not classify it as an affiliation.
- Also to be sure that we do not mark a line in the main text part as affiliation, we check if the line has less than or equal to 3 words starting from lowercase or not. If not, it is not marked as affiliation.

2.3 Author-Email Mapping

OCR++ is also designed to map authors' name with their respective e-mail ids. The challenge faced while performing this task was that different people write their email ids in different ways.

The two most common ways are:

- First name, Middle name or Last Name is a substring of the email.
- The email is an abbreviated form of the person's full name.

Also, in most research papers it is often found that the order of occurrence of emails is same as the order of occurrence of the respective authors. Our algorithm maintained an array of the extracted emails and author names for a research paper and tried to match every author's name with a corresponding email. If a match was found, it was noted in a file and the email was removed from the array. For mapping, first all the names were tried to match using the substring rule. If any email was left, it was matched using the abbreviation rule. This also made sure, the author names were matched to email ids in respective ordering.

2.4 Headings and Section Mapping

For section mapping, we have decided to employ a simple heuristic to first segment the PDF text into regions or **chunks** by means of its distance from neighbouring text and a measure of how different it is from the surrounding text (in terms of font-size, bold-text, etc). Some examples of ideal chunks can be:

- Title of the article/Journal
- Details of the Authors
- Headings/Sub-Headings

- Content of each Section
- Tables/Figures(text only)
- Headers and Footnotes

Table 2:Accuracy results for headings and section mapping task

Format	Precision(%)	Recall(%)	F-Score(%)
acl	90	90	90
acm	90	95	92.43
arxciv	100	67	80.23
ieee	93	90	91.5
Total	94.22	90	92.06

Note that definition for a chunk is rather obscure as it depends on the format of the article, which is quite varying.The criteria for chunking can be summarized as follows:

1. The modal distance between line in the document is calculated. This would be the distance between 2 consecutive lines in a paragraph.
2. We then *chunk* the text based on its modal distance from its neighbours.
3. The chunks derived are further improved upon by rechunking larger chunks if the have a prefix of small(<15) size, which is different from the rest of the text, by font-size.
4. Next, several features are extracted for each chunk, on which a CRF is trained, to get the desired model. Features include:
   - First Token of chunk
   - Second Token
   - Size of chunk
   - Average boldness
   - Normalized Average font-size
   - Features of the first and second tokens viz. a number (Arabic/Roman/ALPHA-enumeration), Special symbol, lowercase token, begins with uppercase letter and others.

The CRF is trained using a template that relates the features to get optimal results like, relating to the chunk sizes of negihbours, relating the features of the first and second tokens, taken along with chunk size, boldness and font-size, etc.

2.5 URL

In order to extract the URL from the scientific papers , we have written a python script which takes the xml file generated from the pdf as input and then matches each and every token with a regular expression .

When the url was inside a pair of parentheses, the parenthesis were stripped and then the token was matched with the regular expression.

Table 3:Accuracy results for extraction of Url

Format	Precision(%)	Recall(%)	F-Score(%)
acl	100	100	100
acm	100	100	100
arxiv	100	100	100
ieee	100	100	100
Total(all)	100	100	100

2.6 Footnote

For the extraction of footnotes, we chose some specific unigram and bigram features that we could conclude by observing some pdf's. Those features were:

- Y-Coordinate
- Size Ratio

The reasons for selecting these features are:

- Y-Coordinate- As footnotes are generally at the end of the page
- Size Ratio- Footnotes are generally small
- Y-Coordinate with size Ratio - Footnotes are generally small and present at the end

Table 4:Accuracy results for footnote extraction

Format	Precision(%)	Recall(%)	F-Score(%)
Total(all)	77.77	27.45	40.57

2.7 Figure and Table Headings

We first employ the heuristic used to segment the PDF text into chunks (described in Headings and Section Mapping).

Table 5:Accuracy results for figure heading extraction task

Format	Precision(%)	Recall(%)	F-Score(%)
acl	100	100	100
acm	100	93.55	96.67
acm_SIG	100	100	100
arxiv	100	100	100
chi	100	100	100
elsevier	100	100	100
ieee	90	94.73	92.20
ieee_journal	100	100	100
springer	100	100	100
Total(all)	98.82	98.24	98.53

Table 6:Accuracy results for table heading extraction task

Format	Precision(%)	Recall(%)	F-Score(%)
acl	100	100	100
acm_SIG	100	83.33	90.99
arxiv	100	100	100
chi	100	100	100
elsevier	100	100	100
ieee	100	71.42	83.33
ieee_journal	100	50	66.67
springer	100	100	100
Total(all)	100	86.79	92.27

- **Figure Headings :** If the chunk starts with the word "FIGURE" or "Figure" or "FIG."or "Fig.", then the chunk represents a figure heading.
- **Table Headings :** It has been observed that the in some cases when the table heading is bold and has same font-size as that of the table contents, then the distance between the heading and the contents of the table are very less, thus causing the contents of the table to be included in the heading.Therefore the chunks derived are further improved upon by re-chunking larger chunks if the chunk starts with "Table" or "TABLE" and the tokens are bold. This is continued till we we observe bold tokens thus excluding the table contents.

2.8 Citations and References

For extraction of references, we use a python script. We extract the title **References** by using features like font-size. The first letter 'R' or the entire word may be in capitals.The word is bold too.
All the tokes succeeding the title "References" are considered to be part of references and each reference is extracted separately.
To extract references we use these features :

- The difference between y-coordinate of the current line and that of previous line.
- Footers and headers are removed using the features like "font" , "font-size" , "bold" and "italic" and "y-coordinate". They will vary from reference to header and footer.
- The starting x-coordinate of lines which are a part of same reference but occur in the next line is more than the first line of the same reference.

We store all these references in a list.

Table 7:Accuracy results for reference extraction task

Format	Precision(%)	Recall(%)	F-Score(%)
acl	67.25	100	79.73
springer	100	96.65	98.3
ieee_journal	100	76.39	86.13
ieee	100	100	100
chi	100	100	100
elsevier	100	100	100
ACM_sig	100	89.87	94.02
arxiv	75	100	83.34
acm_journal	100	97.89	96.79
Total(all)	93.58	94.28	94.70

2.8.1 Citations

For extraction of citations, we use regular expressions. We iterate through the whole document line by line and search for specific regular expressions in the document.

2.8.2 To map citations to references

- *Indexed citations*: Indexed citations are mapped directly to references with an index. Table 8 presents accuracy scores for this task.
- *Non-indexed citations*: In this case author name(s) and year of publication are searched for in the list of references.

Table 8:Accuracy results for citation and reference mapping with index

Format	Precision(%)	Recall(%)	F-Score(%)
ieee_journal	84.78	76.92	80.65
ieee	97.435	100	98.68
chi	100	100	100
arxiv	100	97.5	98.73
ACM_sig	100	97.05	98.5
acm_journal	100	91.83	95.74

Table 9:Accuracy results for citation and reference mapping without index

Format	Precision(%)	Recall(%)	F-Score(%)
acl	58.885	87.755	69.125
springer	58.33	56	57.14
elsevier	92.30	73.46	81.80

3. CONCLUSIONS

We believe that the text processing modules will be a central component of the future digital libraries. The goal of OCR++ is to support various user tasks in relation to large article repositories, in particular the assistance for self archiving of articles, the pre-processing of documents for information retrieval, the generation of reliable OpenURL links or automatic citation suggestions.

4. REFERENCES

[1] J. Beel, B. Gipp, S. Langer, M. Genzmehr, E. Wilde, A. Nürnberger, and J. Pitman. Introducing mr. dlib, a machine-readable digital library. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 463–464. ACM, 2011.

[2] I. G. Councill, C. L. Giles, and M.-Y. Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, 2008.

[3] R. Kern, K. Jack, M. Hristakeva, and M. Granitzer. Teambeam meta-data extraction from scientific literature. *D-Lib Magazine*, 18(7):1, 2012.

[4] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742, 2013.

[5] P. Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Research and Advanced Technology for Digital Libraries*, pages 473–474. Springer, 2009.

[6] C. Ramakrishnan, A. Patnia, E. H. Hovy, G. A. Burns, et al. Layout-aware text extraction from full-text pdf of scientific articles. *Source code for biology and medicine*, 7(1):7, 2012.