Multi-level Information Extraction from Unstructured Scientific Papers

Hao Wang
Shanghai Jiao Tong University
515021910060

ABSTRACT

In this project, the problem of multi-level information extraction is studied. Currently most of our data of papers is only basic information of the paper, e.g. author, title, reference, etc. But for more complicated problems, such basic information are far from enough. And this project aims at using the original information of the paper to do deeper and detailed analysis and information extraction.

KEYWORDS

Information extraction, SVM, CRF.

1 INTRODUCTION

In our Acemap scientific system, there are a lot of networks, e.g., Academic Map, Topic Map. Affiliation Map[7], etc. And most of such networks mainly based on basic information of scientific publications. For example, the academic map is proposed based on the citation relations. And the co-author map considers the author information to finish the visualization.

But there are occasions when such low level information is not enough to solve more complicated problems. Considering that most of our data is crawled from online source, for example, Google Scholar, such data is very limited in our future information analysis. Because what is provided in the online web source is all of what we can get about this paper. To be specific, in name disambiguation, which tries to distinguish authors with the same names, given some basic metadata of paper, like author name, title and reference is not able to help with the disambiguation[6]. And also, for recommendation system of scientific paper, one method is to use the citation networks to do the recommendation. But there is no doubt that different citation has different importance, how to distinguish those less meaningful or incidental citations? We cannot achieve it without knowing about the original paper.

Thus, detailed information analysis based on the original paper is imperative. And in our Acemap dataset, we have the full original document information of scientific papers on a scale of nearly five million. Thus, multi-level information extraction and analysis can be implemented base on these papers.

This project mainly focuses on two targets. Firstly, it tries to extract basic information (title, author, affiliation, email, reference) from PDF files. Secondly with XML of the original PDF, deeper logical analysis and citation filtering can be realized. The basic XML structure labeling is implemented based on CRF (conditional random field). And the further citation filtering is carried on based on SVM (support vector machine) with multiple features. The overall flow chart of the project is show in **Fig. 1**.

This report is organized as follows, firstly, the methodology will be introduced and then comes the implementation of the whole system. Finally, current results will be displayed and also discussion about the current results.

, ,

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

2 METHODOLOGY

In this project, in different steps, different method and model will be used to solve the problem. In general, PDF files are firstly converted to XML files. Then cascading of linear chain CRF (conditional random field) is used to label the unlabeled XML files. Finally, given the labeled XML file, citation filtering can be implemented. An abundance of features is selected to train the SVM as a citation classifier.

And the methodology of PDF to XML, labeling XML and citation filtering will be introduced.

2.1 Reconstructing the structure of scientific papers in XML format

In this part, a rule-based system designed to reconstruct the logical structure of scientific papers in PDF form is introduced, regardless of their formatting style. The system's output is an XML document that describes the input article's logical structure in terms of title, sections, tables, references, etc[2]. But the sections in the article body is unlabeled yet (this will be discussed in the next part).

The key aspect of the presented approach is that the rule set used relies on relative parameters derived from font and layout specifics of each article, rather than on a template-matching paradigm. The system thus obviates the need for domain- or layout-specific tuning or prior training, exploiting only typographical conventions inherent in scientific literature.

In order to address structure recovery, it carries out a two-stage process. The first stage constructs a geometrical model of the article's contents to determine the spatial organisation of textual and graphical units on page. The second stage draws upon the first to identify different logical units of discourse based on their discriminative features.

With the geometrical model and statistics in place, the second stage attempts to determine the semantic roles of the newly created blocks, possibly merging them into logical regions in the process.A sequence of steps aims to identify one logical element type at a time, across the whole article, by tagging regions with certain characteristics.

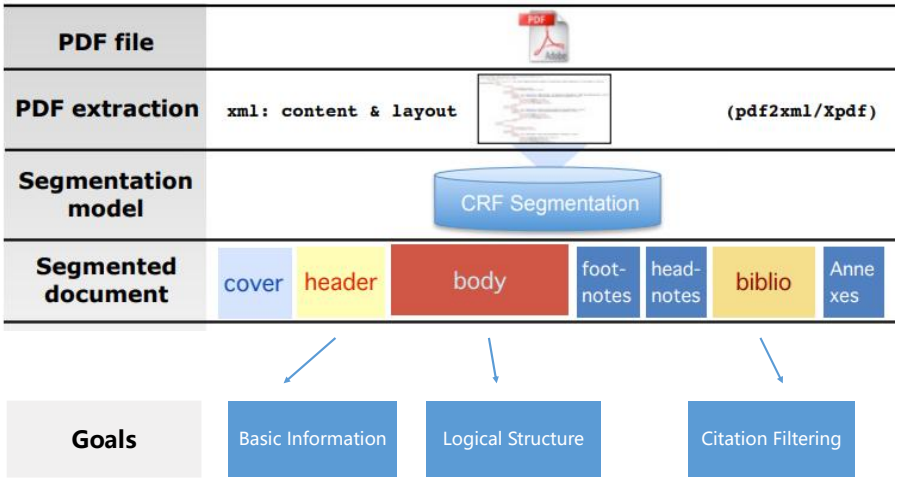

Figure 1: Flow chart of overall goal.

The first and most important step in the sequence is to identify the core body text along with the reading order of the article. Out of the set of merged blocks, those containing primarily words in the most frequent font of the article are tagged as body regions. The dominant body region shape is used to determine the column layout and the intended reading order. Tagging of the rest of the regions is afterwards carried out in a prioritised manner. The priority is dictated by an empirically determined level of difficulty in identifying each logical element type. The elements considered easiest to tag confidently are searched for first. The rest of the identification sequence is as follows: (1) images, (2) DOI, (3) authors, (4) title, (5) outsiders: headers, footers, side notes, page numbers, (6) toplevel headings, (7) abstract, (8) captions, (9) lower-level headings, (10) author footnotes, (11) remaining regions, (12) bibliography and bibliographic items, (13) other body regions, (14) tables, (15) in-text references to figures, tables and bibliographic items; URIs and emails.

2.2 Labeling XML file to indicate logical structure of scientific papers

In this part, an approach[3] based on cascading of linear chain CRF will be introduced to label the data of XML files. And in implemention, 11 CRF models will be trained to be used for full text information labeling, which is a coarse level segmentation, and for refined level, header information labeling, body information labeling, reference information labeling, etc. Since the methodology of different CRF models is the same, for simplicity, the CRF model of segementing the whole XML file (coarse level) will be introduced here.

2.2.1 Conditional Random Filed. Conditional random fields (CRFs) are one kind of generative model and a class of statistical modeling method used for structured prediction. CRFs fall into the sequence modeling family. Whereas a discrete classifier predicts a label for a single sample without considering "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF (which is used in this model) predicts sequences of labels for sequences of input samples[4].

For an observed sequence $O = (o_1, ..., o_2)$, given a sequence $I_k = (i_1, ..., i_i)$. Let

$$P(I_k \mid O) = \frac{1}{Z(O)} e^{\sum_i^T \sum_k^M \lambda_k f_k(O, I_{i-1}, I_i, i)}$$

where $P(I_k \mid O)$ denotes the probability of O given I_k as the sequence of hidden state. f_k is the $ch.f$ and λ_k is the weight for corresponding f_k. In each cluster, there are in total M features. $Z(O)$ is used for normalization.And of all $I_k(k = 1, 2, 3)$, one with the largest $P(I \mid O)$ will be selected as the labeled sequence for O.

2.2.2 Apply CRF to XML. The flow chart of labeling (segmentation) XML file is shown in **Fig. 2**. In this chart, after getting the XML file of the original PDF file, a CRF model is used for coarser level XML segmentation. In other words, divide the XML file into cover, header, body, footnotes, headnotes, biblio, Annexes, if there is. We need to notice that part of the work can also be done using method in the previous methodology part. And then with header, body, biblio, etc XML file, corresponding CRF model will be used to do a refined level segmentation, e.g., header will be segmented into title, authors, affiliations and so on (see **Fig. 3**). And in this way, different part of XML file has been successfully labeled into our desired sequence.

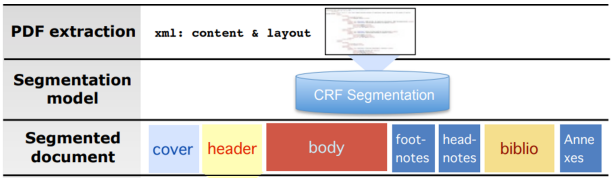

Figure 2: Flow chart of labeling (segmenting) XML file.

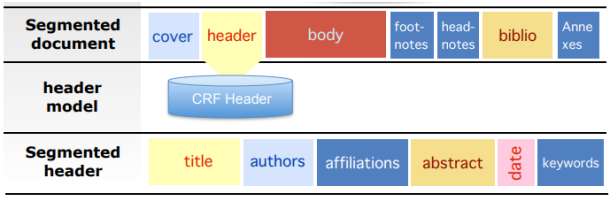

Figure 3: A refined level of labeling the header information of document.

2.3 Citation filtering

Citation filtering is used for finding important citations where the word important is driven by three key observations:

- The more citations a paper receives in the body of the citing work, the more important the citation is likely to be.
- It matters where the citation appears. For example, a citation in the Related Work section is likely to indicate an incidental citation. On the other hand, a citation in the Methods section indicates that the cited work is used or extended in the citing paper, which signals importance.

Before carrying on, citations also appear in many forms. Some are *direct*, i.e., the citation follows an established proceedings format, or *indirect*, where the work is cited by mentioning the name of an author, typically the first author, the name of the cited algorithm, of a description of the algorithm. Thus, in order to reliably implement the two observations, one has to identify both direct and indirect citations.

Based on the above observation, we can futher derive our model based on the following 12 feature set[5].

- **(F1) Total number of direct citations**: This feature counts the total number of citations to the cited paper.
- **(F2) Number of direct citations per section**: Similar to the above feature, but counts are qualified by the section in which they appear. For this feature we used the normalized section titles produced by ParsCit. For example, if a paper has five citations, with two appearing in the Related Work section and three in Methods, we generate two features: DirectCountsRelatedWork with a value of 2 and DirectCountsMethods with a value of 3.
- **(F3) Total number of indirect citations and number of indirect citations per section**: Similar to the previous two features but focusing on indirect features. Since the description n-grams may be redundant (i.e., we may find multiple, slightly different descriptions of the same work) we count them differently than direct citations: instead of counting occurrences, we count the number of sentences in which at least one potential description appears.
- **(F4) Author overlap (Boolean)**: This feature is set to true if the citing and the cited works share at least one common author. The intuition behind this feature is that shared authors indicate that the new work is likely to be an extension of the cited paper.
- **(F5) Is considered helpful (Boolean)**: This feature is set to true if a sentence in which a citation occurs contains phrases such as "we follow" or "we used", which are hints that the author of the citing work considers the cited paper to be important.
- **(F6) Citation appears in table or caption (Boolean)**: Set to true if at least a citation appears in a table or a caption of a figure or table. This is an indicator that the author of the citing work is comparing her results to the cited paper.
- **(F7)1 / number of references**: This feature computes the inverse of the length of the citing paper's reference list, which hints to the value of receiving one citation, e.g., if it is one citation from a total of two references, this citation is clearly important.
- **(F8)Number of paper citations / all citations**: Similarly, this feature computes the number of direct citations instances for the cited paper over all the direct citation instances in the citing work.
- **(F9)Similarity between abstracts**: This feature computes the similarity between the cited and citing paper's abstracts using the cosine similarity of the tf-idf scores. The intuition behind this feature is that the closer the abstracts, the more likely the new work extends the cited paper.
- **(F10) PageRank**: This feature computes the PageRank score (Page et al. 1999) of the cited paper, as a measure of the cited work's importance.
- **(F11) Number of total citing papers after transitive closure**: This feature records the number of citing papers after the transitive closure, e.g., papers that cite the cited work, papers that cite those papers, etc.
- **(F12) Field of the cited paper**: This feature stores the particular computer science subfield to which the cited paper belongs. This is work in progress: we currently developed a classifier that identifies if a paper describes a software system or not. This classifier was developed as part of a scientific literature search engine and is based on bag of words technique matching system names with citation contexts.

Thus based on these features, a SVM is trained to finish the classification task.

3 IMPLEMENTATION

The implementation of PDF to XML conversion is based on the rule-based system mentioned in methodology part. The XML output is constructed with the most likely tags of the different regions at the end of the processing sequence. The initially identified contiguous blocks, now encapsulated in logical regions,jointly reconstruct the rhetorical structure of the article. Information about the different regions and their organisation is represented using an XML format

very close in schema to the JATS standard. The logical section elements are implied by the heading hierarchy, being added in and populated during the XML construction. As regions can span multiple blocks, columns or pages, their respective XML elements may contain tags that act as physical position markers in the original text. An excerpt from the processing output of this paper illustrates a region spanning two pages (See **Fig. 4**). The intruding figure itself was identified and skipped over when reconstructing the text stream. The class attribute of the region, set in accordance with DoCO[1], was added in order to facilitate interoperability with other services. DoCO is an ontology of both physical and logical components of bibliographic documents, wellsuited for linking PDFX output to other text processing pipelines.

```
<region class="TextChunk" page="2" column="2">
    [...] encapsulated in logical regions,
    <marker type="page" number="3"/>
    <marker type="column" number="1"/>
    <marker type="block"/>
    jointly reconstruct the rhetorical [...]
</region>
```

Figure 4: Example of a region spanning two columns.

The implementation of labeling XML file is in fact the training of CRF models. The traning process is carried on in a recursive manner. When the current model is fed with a new document, this model is going to label this XML file and output the labeled result, after human correction, this is going back to update the current CRF model. And then after enough iterations we can get the final CRF model and stop the training. **Fig.5** shows the recursive process.

And for each model the same generic CRF-based framework is used which covers training, evaluation, tokenization, decoding, etc. And the different set of features and training data and normalization are used. The features mainly consider the following information:

- Position imformation: begin/end of line, in which part of the doc, etc.
- Lexical information: vocabulary, large gazetteers, etc.
- Layout information: font size, block, font style, etc.

The implementation of citation filtering is simply based on sklearn toolkit and use the SVM to train the classifier.

4 RESULTS

The rule-based PDF to XML conversion has been successfully applied to a small dataset (around one thousand pdfs) and the outcome is shown in **Tab. 1**. The reason why the precision of Header is relative lower than recall is that sometimes this system misdivides the full text to the header part leaving the rest of the XML sections empty.

Table 1: result of PDF to XML conversion

Category	Precision	Recall
Header	82.4%	93.4%
Section	90.1%	81.1%
Subsection	84.2%	73.7%
Tail	95.2%	89.3%

Table 2: result of labeling the header information of XML

Category	Precision	Recall
Title	93.8%	88.1%
Authors	85.4%	82.1%
First Author	96.1%	89.7%
Abstract	84.3%	78.6%
Keywords	87.2%	70.5%
Email	94.2%	98.3%

This outcome shows that such rule-based system can successfully achieve the conversion especially for header and section extraction. This result paves the way for further applying it to Acemap dataset which is on a million scale.

The results of labeling XML is shown in **Tab. 2**. At current stage, only the header information is tested. Since the XML conversion is still being improved for a better result. Considering that the conversion may have some minor mistakes when converting text of PDF to XML, the statistics in the table are under a measure standard which allows the labeled result(lr) and ground truth(gt):

$$\frac{Levenshtein\ distance(lr, gt)}{length(gt)} < 0.1$$

.

where lr and gt are strings of the results. From the table, we can draw the conclusion that at current stage, the basic information extraction from PDF can be applied to our larger Acemap dataset. And with days efforts running on the server, basic information from header for nearly five million papers has been successfully ectracted. Currently improving the results of XML conversion for body part is still ongoing.

And for citation filtering, with labelled dataset of nearly 100 papers, a SVM is trained using cross validation. With a precision of about 65% and recall of about 78% (still ongoing). we found that, those citations classified as 'important' mainly appear in experiment-like or methodology-like part while those appearing in related work or introduction are more likely to be less important.

5 CONCLUSION

In this project, the problem of multi-level information extraction is studied. We can take advantage of abundant dataset in Acemap dataset to accomplish multi-level information extraction. And the outcome shows that for basic level information extraction, the method proposed in the paper has been successfully applied to all the data in Acemap dataset and achieves impressive result. The XML conversion of body has achieve a promising result but is still being improved to applied it to larger scale. Based on linear chain

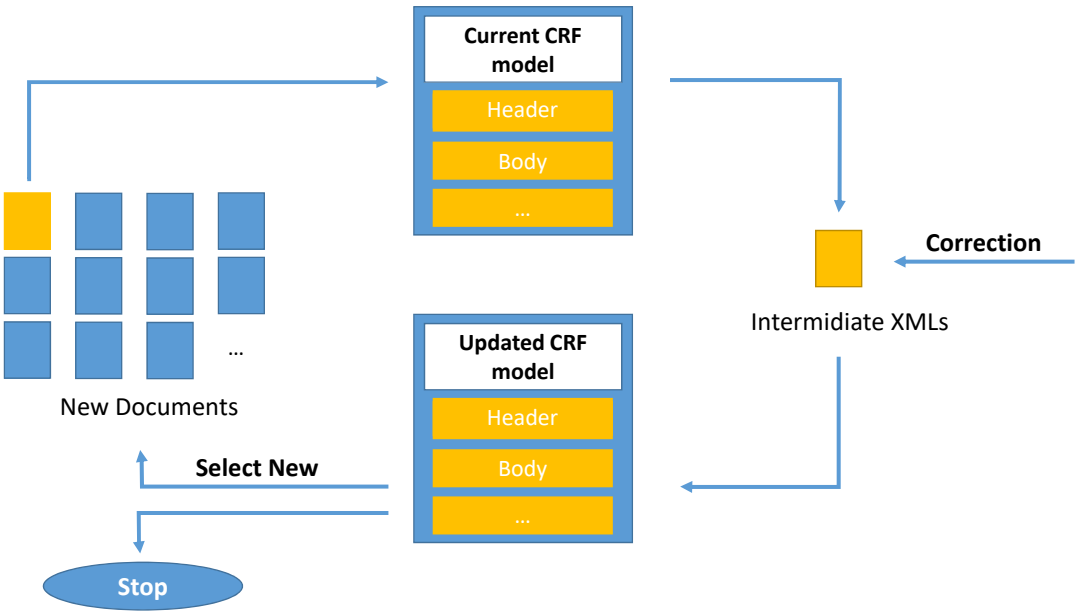

Figure 5: Recursive process of training the CRF model (take the coarse level CRF model as an example).

CRF model the labeling for header information can be labeled. And the citation filtering is still under research to improve the filtering result.

REFERENCES

[1] Doco. http:www.purl.org/spar/doco.
[2] Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. Pdfx: fully-automated pdf-to-xml conversion of scientific literature.
[3] P. Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. 2009.
[4] A. McCallum. Efficiently inducing features of conditional random fields.
[5] Marco Valenzuela. Identifying meaningful citations. 2015.
[6] Dina Vishnyakova, Raul RodriguezEsteban, Khan Ozol, and Fabio Rinald. Author name disambiguation in medline based on journal descriptors and semantic types. 2016.
[7] Xinbing Wang. Acemap. http://acemap.sjtu.edu.cn.