Information Extraction from Research Papers by Data Integration and Data Validation from Multiple Header Extraction Sources

Ozair Saleem, Seemab Latif

***Abstract*—Massive amount of information is available on the web in form of Research paper publications. Extracting Header information like Conference Name, Title, Authors, Affiliation, Email, Keywords and Abstract can be very useful in performing data mining tasks like finding research trends in particular research area or finding collaboration done among different research groups or universities. Existing Header Parser tools identified by Yao et.,al.[1] includes GROBID, ParsCit, Mendeley, HeaderParserService, PDFSSA4MET, PDFMEAT, Zotero and PaperPile. Tools using Machine Learning Algorithms include GROBID, ParsCit, Header Parser Service and Mendeley. Now the problem faced here is that no single tool gives 100% results against all sample research papers. One tool outperforms other in identifying individual elements. For this reason one cannot rely on single tool for all elements extraction. In this paper, we are proposing a hybrid method for the extraction of header information from the papers using GROBID, ParsCit and Mendeley. Results of these tools are merged to achieve accurate header extraction. This proposed method has been applied on 75 sample research papers and the overall accuracy of 95.97% is achieved.**

***Index Terms*— Information Extraction, Header Extraction, Data Pre Processing.**

I. Introduction

Research is an on-going process which results in writing Research publications that are shared with the world in different conferences. Portable Document format or what we usually call PDF is the usually the mechanism used to share Research publications. Research publication header information that includes Conference name in which paper was published, Title of paper, Author(s) of paper, Affiliations of author(s), Email of Author(s), Keywords in paper and Abstract of paper. This information when extracted can be very useful in many data mining scenarios for example one can find collaboration among different universities by looking for publication which have authors from two or more than two different universities. Another scenario can be one can find trends of research done in particular field or area.

The typical problem faced in extracting information from research papers is that they do not have a common format defined in which research publications are written. Usually every conference or journal has its own format for writing research publication.

Information extraction from research paper is done by three approaches. First approach is to do structural analysis of PDF along with pattern matching. For example, matching for words like Abstract and Keyword and looking for font size to extract Title as title is usually of larger font size. This approach overall is not very efficient due to large number of paper structural formats in which papers are published. Another approach used is Web based lookup from a knowledge base. In this Approach one element of paper for example, Title is extracted and then web based lookup is done from an online knowledge base. Last approach commonly used for extracting header information is by training machine learning algorithms to identify elements in paper. Machine Learning Techniques used are Support Vector machine (SVM)[3], Hidden Markov Model (HMM)[4] and Conditional Random Field (CRF)[2]. Header extraction tools that were identified by Yao et.,al.[1] that are using machine learning tools are GROBID and ParsCit are using CRF, Header Parser Service and Mendeley relying on SVM.

II. Related Research

Existing Header Parser tools identified by Yao et.,al.[1] includes GROBID, ParsCit, Mendeley, HeaderParserService, PDFSSA4MET, PDFMEAT, Zotero and PaperPile.

Generation of Bibliographic Data(GROBID)[5] has used Conditional Random Fields machine learning algorithm that is implemented using MALLET[12] to extracts the bibliographical data corresponding to the header information (title, author, affiliation, email, keyword and abstract) and references(title, authors, journal title, issue, number).

Parse Citation (ParsCit)[6] performs Reference string parsing and logical structure parsing of scientific documents. ParsCit uses Conditional Random Fields as its learning mechanism.

Mendeley[7] uses Support Vector Machines and Web based lookup for Extraction of embedded metadata and extraction of citation details from research publication.

Manuscript received July 2, 2012; revised August 5, 2012.

Ozair Saleem is student in the Department of Computer Software Engineering, College of Telecommunication Engineering, National University of Sciences and Technology(NUST) Islamabad, Pakistan (phone: +92 333 4622909; e-mail: ozair.saleem@gmail.com).

Seemab Latif is with the Department of Computer Software Engineering, College of Telecommunication Engineering, National University of Sciences and Technology(NUST) Islamabad, Pakistan (e-mail: seemab@mcs.edu.pk).

Mendeley provides windows based application that helps in organizing and collaboration of research publication. When research publication is added, Mendeley identifies the Author, Title, Journal, Year and keywords.

PDF Structure and Syntactic Analysis for Metadata Extraction and Tagging[8](PDFSSA4MET) provide metadata extraction and tagging based on structural and syntactic analysis of research publication. PDFFSSA4MET extracts and tag Title, Author, Section headings and references.

PDFMEAT[9], Zotero[10] and PaperPile[11] do web based lookup for getting Research publication information.

Yao et.,al[1] proposed a framework that will call PDFSSA4MET, ParsCit, HeaderParserService and PDFMEAT from command line and generate a uniform result set from it.

III. Pre Processor

The proposed hybrid approach in this paper is different from Yao et.,al[1] in a way that this approach will not be calling GROBID, ParsCit and Mendeley directly from code, Instead XML outputs will be taken by each of these tools and then they will be further processed to produce more accurate results. GROBID and ParsCit generate XML file for each research publication. Mendeley gives one XML file containing list of all research publications. The architecture of proposed system is in Figure 1.

Initially each tool is used separately to pass research publication and get XML output for each publication. Then these XML files are collected for each tool and passed to the XML Reader modules. XML Reader module parses individual elements from XML files and stores them to relevant Classes of GROBID, ParsCit and Mendeley respectively. XML reader identifies same research publication from GROBID, ParsCit and Mendeley XML file based on the file names of publication. Then for each publication parsed elements from GROBID, ParsCit and Mendeley are sent to Pre Processor. Pre Processor primarily does two tasks that are data integration and data validation. Final output from Pre Processor module is computed and stored in Publication Database.

A. *Criteria for identifying Correct Data*

Once header results have been read from 3 sources major challenge is defining the criteria the correct result from incorrect one.

The approach that we will be using in this paper is first to check for exact match of string form all the sources. If there is an exact match then header element is set to one of the header results.

Exact_Match(GROBID.Title , ParsCit.Title , Mendeley.Title)

If there is not the exact match found then we check for percentage of match between sources. A threshold is defined like if results from 2 or 3 header tools are 90% matching then they are correct results. This approach solves problem when one of the Header Extraction source completely fails to extract an element of paper.

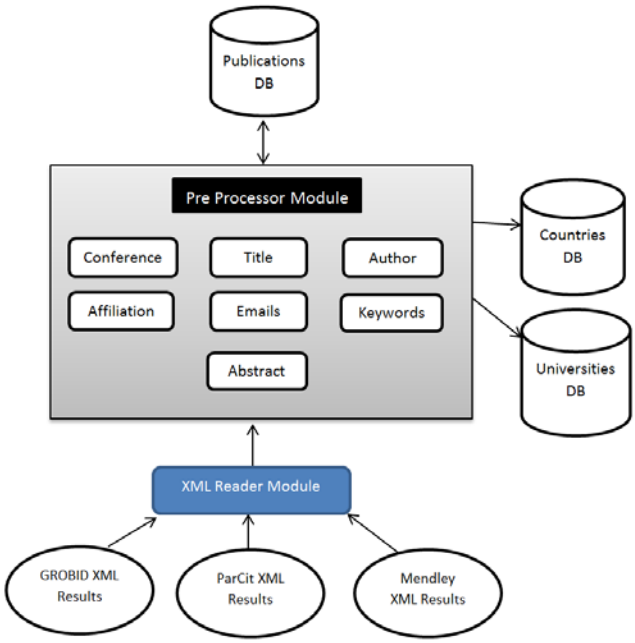

Fig. 1. Architecture of Proposed System

Percentage_Match(GROBID.Title , ParsCit.Title)

For some header elements like finding affiliation and country, we have defined database of all possible affiliation in the world. There is another database defined for all possible countries in the world. For verification of country and affiliation a search is done from the database.

Search_Affliliation_DB(GROBID.Author.Affiliation)

We have defined two confidence levels. If an element is validated from 2 or 3 sources its confidence become TRUE otherwise it confidence will be FALSE that means it is confidence of extraction from one source only. Publication items with Confidence of TRUE are always assumed to be correct.

B. *Problems faced with existing header results*

There were many problem identified when GROBID, ParsCit and Mendeley are run for different kind of publication. They are listed below:

- ParsCit mixes Conference with title in title XML tag.
- ParsCit title appears in note instead of title XML tag.
- ParsCit fails to separate Authors if they are not separated by commas.
- ParsCit Abstract mixes with other section of research paper like introduction when Research paper is in two column format.
- ParsCit combines keywords with other section of research paper when the paper is in two column format.
- ParsCit keywords are embedded in Abstract XML tag.
- ParsCit Emails are not separated by spaces. ParsCit Email appears in Address XML tag.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- GROBID maps many affiliations to one author.
- GROBID do not map any affiliation to any author.
- GROBID fails to identify Title field.
- GROBID fails to extract Authors.
- GROBID fails extract Affiliations.
  GROBID abstract also contain keywords embedded in XML tag.
- Mendeley miss some Authors.
- Mendeley extracts incorrect keywords.

IV. Algorithm For Components Of Pre Processor

We have identified the criteria for correct data and also identified common problem that occur with existing results of GROBID, ParsCit and Mendeley. For all pseudo code written, it's assumed that header results were extracted from all three header extraction sources. There is however, possibility that individual elements like title, abstract is not extracted by individual header extraction source. We will go one by one to individual elements of Pre-processor solving the identified problems.

A. Title Module Pseudo Code

Title field is extracted form GROBID, ParsCit and Mendeley. Here is the Pseudo code.

```
Confidence = false
if(Exact_Match(GROBID.Title , ParsCit.Title , Mendeley.Title)
{
   Add title to final publication
   Confidence =true
}
Else if(Percentage_Match (GROBID.Title , ParsCit.Title ,
Mendeley.Title)
{
   Add title to final publication
   Confidence =true
}
Else Check all combination of 2 sources
{
  if(Exact_Match(Source1.Title , Source2.Title)
  {
     Add title to final publication
     Confidence =true
  }
  Else if(Percentage_Match (Source1.Title , Source2.Title)
  {
     Add title to final publication
     Confidence =true
  }
}
Else Add title extracted from 1 Source only to final publication
```

B. Conference Module Pseudo Code

Conference name are extracted by ParsCit and Mendeley only. In ParsCit Conference appears in the Note XML tag or it is embedded in Title XML tag. In Mendeley Conference name appears in the Secondary title XML tag but that is not always true because for some cases Domains like Image Processing appear in secondary title. Here is the pseudo code.

```
Confidence = false
if(ParsCit Title contains text of Mendeley Secondary Title )
{
   Add Mendeley Secondary Title as Conference Name
  Confidence = true
}
Else if(Percentage_Match (ParsCit.Title,Mendeley.Title )<50)
{
  Remove string of Mendeley title from ParsCit Title and Add as
Conference Name
  Confidence = true
}
Else if(ParsCit Note contains text of Mendeley Secondary Title )
{
   Add Mendeley Secondary Title as Conference Name
  Confidence = true
}
Else if(Mendeley Secondary Title Length is great than 50 )//this
will avoid assigning domains to conference
{
   Add Mendeley Secondary Title as Conference Name
  Confidence = true
}
Else there was no conference extracted
```

C. Author, Affiliation, Email and Address Module Pseudo Code

Authors are extracted for all sources. Affiliation, Email and Address are extracted from GROBID and ParsCit only. This Module is the most difficult one to extract correctly as Affiliation are mapped to an Author, Email is mapped to an Author and Address is mapped to Affiliation. Only GROBID already have these fields mapped, ParsCit only identifies them. Here is the Pseudo code.

```
if( Extracted from GROBID ParsCit and Mendeley)
{
  • Verify that affiliation by searching from AffiliationDB
  • Verify that Address by searching from CountriesDB
  • Validate GROBID email with ParsCit email
  • Validate GROBID authors with ParsCit and Mendely
      Authors
  • Remove Extra Affiliation from GROBID authors
  • Map extra affiliations to authors with no affiliation
  • Map GROBID Authors with missing email by using
      Levenshtein Distance algorithm
}
ELSE if( Extracted from ParsCit and Mendeley)
{
  • Verify that affiliation by searching from AffiliationDB
  • Verify that Address by searching from CountriesDB
  • Map emails to authors by using Levenshtein Distance
      algorithm
  • Map Affiliation to authors

}
 ELSE if( Extracted Mendeley only)
{
  • Add only Authors to final publication
}
```

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

D. Abstract Module Pseudo Code

Abstract is extracted from GROBID and ParsCit only. Here is the pseudo code for getting Abstract.

```
Confidence = false
if(GROBID Abstract contains Keywords)
{
  Remove Keywords from GROBID Abstract
}
if(ParsCit Abstract contains Keywords)
{
  Remove Keywords from ParsCit Abstract
}
if(Exact_Match(GROBID.Abstract, ParsCit.Abstract)
{
   Add Abstract to final publication
  Confidence = true
}
Else if(Percentage_Match(GROBID.Abstract, ParsCit.Abstract)
{
   Add Abstract to final publication
  Confidence = true
}
Else if(Abstract read from GROBID Only)
{
   Add Abstract to final publication
}
Else if(Abstract read from ParsCit Only)
{
   Add Abstract to final publication
}
```

E. Keyword Module Pseudo Code

Keywords are extracted from GROBID, Mendeley and ParsCit. It was observer that GROBID was most accurate in extraction of keywords. For this reason if keywords are extracted from GROBID only and then ParsCit and Mendeley are only used to validate the keywords. However if keyword extraction from GROBID fails then keywords are added from ParsCit and Mendeley. Here is the pseudo code.

```
Confidence = false
AreKeywordsPresent=false
if(GROBID contains keywords)
{
   Add Keywords to final Publication
  AreKeywordsPresent=true
}
Else if(ParsCit contains keywords &
AreKeywordsPresent==false)
{
   Add Keywords to final Publication
}
Else if(ParsCit contains keywords &
AreKeywordsPresent==true)
{
   Validate existing final Publication
  Confidence = True
}
Else if(Mendeley contains keywords &
AreKeywordsPresent==false)
{
   Add Keywords to final Publication
}
Else if(Mendeley contains keywords &
AreKeywordsPresent==true)
{
   Validate existing final Publication
  Confidence = True
}
```

F. Year Module Pseudo Code

Year Field is only extracted from Mendeley. GROBID ID No contains year along with other digits. Here is the pseudo code for getting Year of Publication.

```
Confidence = False
if(Mendeley contains Year field)
{
   If(GROBID ID NO Contains Mendeley Year)
     {
      Add Year to final Publication
      Confidence = True
      }
  Else
      Add Year to final Publication
}
```

V. Results

This algorithm is tested on a small set of 75 Research Papers with an overall element extraction accuracy of 95.97%. It can be seen from the table that results of Pre-processor have improved for all extracted elements except Year field.

The efficiency of the proposed Pre Processing Technique depends only on the outputs of the GROBID, ParsCit and Mendeley. If none of them are able to extract an element of paper then the Pre-processor will not be able to extract that element either. However, if one or two of them have failed to extract an element Pre-processor will be extracting that element correctly.

Another problem faced when mapping extracted affiliation is that Affiliation is not present in the Affiliation Database or present with a slightly different name or abbreviation of affiliation is used instead of full name for example MIT is used for *Massachusetts Institute of Technology*.

TABLE I. Results

	Mendeley	ParsCit	GROBID	Pre-processor
Title	97.57	76.80	95.23	99.06
Conference	27.27	75.38	N/A	92.18
Author	92.81	64.81	83.47	94.15
Affiliation	N/A	73.07	86.92	95.42
Email	N/A	89.61	85.49	97.90
Abstract	N/A	82.23	98.15	98.44
Key Word	81.51	83.30	89.23	98.43
Year	92.42	N/A	10.77	92.42

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

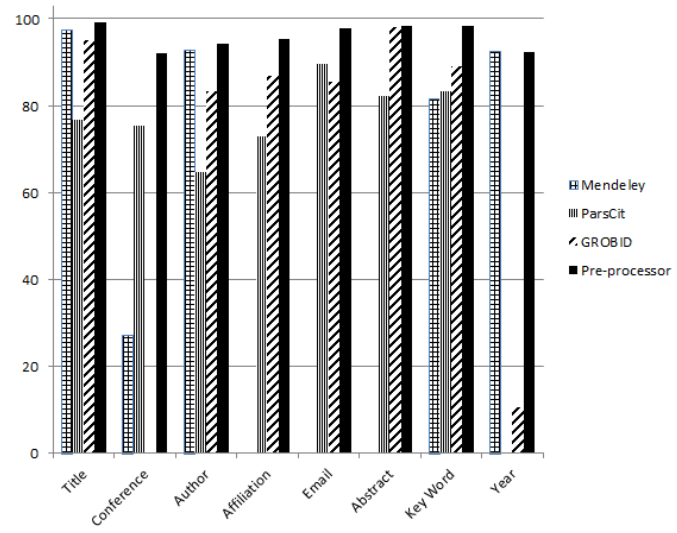

Fig. 2. Results Graph Comparison

VI. Future Work

First problem can be solved by adding a module that will be doing web based lookup for the publication. Microsoft Academic Search provides access to records of 38 million publications and 19 million authors. Microsoft Academic Search provides a web service that can be used to access records of different universities, conferences, publication, authors and academic domains. An element can be extracted from the header extraction source and then use that element to search Microsoft Academic Search database. For example, if all the three Header Extraction sources were failed to find Authors of a paper then find the Title of the paper and search that paper in Microsoft Academic Search to get record of the publication hence improving further accuracy of a publication document.

Second problem can be solved by creating a better database of all universities along with their abbreviations and slightly different names.

VII. Conclusion

In this paper we presented a hybrid approach for extracting header information from Research Papers. This technique uses Data Validation and Data Integration from three Header extraction sources to produce better results.

References

[1] Kevin Yao, Mario Lipinski, Bela Gipp and Jim Pitman. “Header Extraction from Scientific Documents”.
[2] Fuchun Peng and Andrew McCallum. 2004. “Accurate Information Extraction from Research Papers using Conditional Random Fields”.
[3] Hui Han, C. Lee Giles, Eren Manavoglu and Hongyuan Zha. 2003. “Automatic Document Metadata Extraction using Support Vector Machines”.
[4] K. Seymore, A. McCallum, and R. Rosenfeld. 1999. “Learning hidden Markov model structure for information extraction”.
[5] Patrice Lopez, “GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction For Scholarship Publications”.
[6] Isaac G. Councill, C. Lee Giles and Min-Yen Kan, “ParsCit: An open-source CRF reference string parsing package”.
[7] Mendeley is a desktop and web program for managing and sharing research papers, discovering research data and collaborating online. Available: http://www.mendeley.com/
[8] PDFSSA4MET attempts to provide metadata extraction and tagging based on structural and syntactic analysis of content in XML. Available: http://code.google.com/p/pdfssa4met/
[9] PDFMEAT is Metadata acquisition and embedding tool for papers in PDF format. Available: http://code.google.com/p/pdfmeat/
[10] Zotero is a powerful, easy-to-use research tool that helps to gather, organize, and analyze sources and then share the results of research. Available: http://www.zotero.org/
[11] Paperpile is a tool to find, organize, cite and share scientific papers. Available: http://paperpile.com/
[12] A. McCallum and A. Kachites.2002. MALLET: “A Machine Learning for Language Toolkit”.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)