

# A New Approach towards Bibliographic Reference Identification, Parsing and Inline Citation Matching

Deepank Gupta<sup>1</sup>, Bob Morris<sup>2</sup>, Terry Catapano<sup>3</sup>, and Guido Sautter<sup>4</sup>

<sup>1</sup> Netaji Subhas Institute of Technology, Plazi

<sup>2</sup> University of Massachusetts, Boston, Plazi

<sup>3</sup> Columbia University, Plazi

<sup>4</sup> University of Karlsruhe, Plazi

**Abstract.** A number of algorithms and approaches have been proposed towards the problem of scanning and digitizing research papers. We can classify work done in the past into three major approaches: regular expression based heuristics, learning based algorithm and knowledge based systems. Our findings point to the inadequacy of existing open-source solutions such as Paracite for papers with “micro-citations” in various European Languages. This paper describes the work done as part of the Google Summer of Code 2008 using a combination of regular-expression based heuristics and knowledge-based systems to develop a system which matches inline citations to their corresponding bibliographic references and identifies and extracts metadata from references. The description, implementation and results of our approach have been presented here. Our approach enhances the accuracy and provides better recognition rates.

**Keywords:** Bibliographic Reference Parsing, Inline Citation Matching, Regular Expression, Metadata Extraction, Knowledge-based Systems, Micro-citations.

## 1 Introduction

Scientific research never happens in a vacuum and always builds upon previous work. Thus, we say that a scientist always stands on the shoulders of Giants. The previous work upon which research is done is cited through references in scientific papers and journals.

This paper describes automatic recognition, parsing and normalization of bibliographic references to enable easy search and retrieval of related information content. Since the fields of scholarship are spread out with work taken up in different countries in different times there has never been a single rigid method of referencing. So, the study of legacy methods of referencing must be carried out and tools must be provided which will be able to extract the information from citations so that it may be utilized irrespective of the format of referencing followed by various journals.

As in most scholarly publications, papers in the field of biological taxonomy contain many inline citations that are severely abbreviated. From an information processing point of view, these are actually references to a complete entry in a

bibliography elsewhere in the paper. The nature of the abbreviation, and the possible requirement to locate and parse the full reference, can make these citations difficult to identify, parse, and extract information from. Taxonomists sometimes call these "micro-citations", although they go by many names in published style guides. Since, we are doing computerized reading of papers, we often do not get reliable information from formatting and thus the formatting cues have to be largely ignored. Apart from this, the separators are semantically overloaded with information and often serve more than one purpose.

In this paper, we will look at a unique combination of information obtained from separators, domain-specific knowledge and localized knowledge of a paper; to obtain good accuracies in the field of parsing and normalizing references.

## 2 Previous Works

The problem of digitizing and categorizing the world's information is not new. It has been studied by many scientists over the last decade or so. A number of suitable algorithms and approaches have been proposed towards the problem of scanning and digitizing research papers, extracting references, extracting metadata from references and citation matching. Some of the notable projects which address the problems presented above are ParaCite, CiteSeer and Google Scholar.

We can classify the work done in the past into three major approaches. Firstly, the regular-expression based heuristics as discussed in [1], [9] have been applied to extract metadata from references. ParaCite Toolkit [1] which is a collection of perl modules often termed as ParaTools uses a standard set of templates to extract metadata from the references. The technique was further perfected by the application of a protein sequence analyzer namely BLAST [9] to generate a more comprehensive set of around 2500 templates against which the citations were matched and indexed. Another novel application of a regular expression based parser has also been developed in [10] to find references between Dutch Laws.

The second approach is to apply learning based algorithms such as Hidden Markov Model, Conditional Random Fields and Support Vector Machines which utilize machine learning and get better with more training data. Hidden-Markov Model [7, 8] is a probabilistic model in which there is transition between a finite set of states accompanied by a corresponding output usually as a symbol from the character set. They are known as hidden as the output is known and the task is to determine the sequence of states through which the model goes to result in the emission sequence.

Lastly, knowledge based systems proposed by Giufridda [2] reports a good accuracy from a narrowed down corpus of computer science based research papers. He uses the spacial/visual knowledge principle for extracting metadata from scientific papers stored as PostScript files. Another interesting work on reference and citation extraction was done by Brett Powley and Robert Dale [3] in which localized information was used from the research paper to identify and match citations inside it.

The three different approaches have all given different results. This paper describes a combination of the conventional approaches of regular-expression based heuristics and knowledge-based systems to develop a system which converts the input files into a computer-readable schema; identification and extraction of metadata from references

and citation matching. This paper has been divided into sections as described here. The General Approach and algorithms have been described in the Section 3. This section will describe the approaches followed towards various problems in the various stages. This will be followed by the results obtained with the test corpus in Section 4 and related observations. Section 5 focuses on the conclusions and the scope of future work regarding the same.

### 3 Terms and Notations

The following terms will be frequently used through-out the paper:

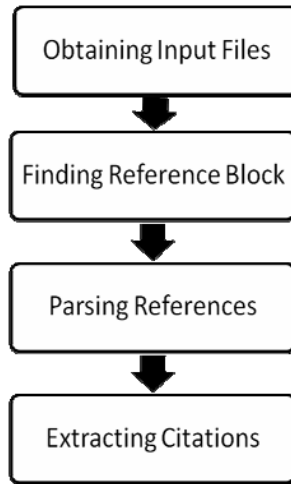
- **Reference:** A Reference appears in the list of works containing the full bibliographic information such as ‘Author Name(s)’, ‘Year of Publication’, ‘Title of Work’ and ‘Journal Name’ about a cited work.
- **Citation:** Present inside the text of the document and contains enough information to identify reference uniquely from the list of references in the document.
- **Inline-textual Citation:** This type of citation is usually a part of sentence and contains an Author, Year Pair to uniquely identify the reference from the list of references in the document.
- **Separator:** It refers to the special character used to separate two different fields of a reference
- **Reference Block:** The section of a document which contains the list of bibliographic references referred to in the document is known as a reference block.
- **Keywords:** These are the special words which mark the start of a reference block e.g. “References” marks the start of the reference block of this document.

### 4 Our Work

The approach and system employed has been described in this section of the paper. The process can be divided into 4 stages as described in diagram 1: Obtaining Input Files in the TaxonX Schema [5] from the scanned documents; obtaining a reference block in the document and identifying references; parsing the references into Author Name, Year, Title, Publication and other metadata; and, identification and matching of the corresponding citations with the references in the document.

#### 4.1 Input Files

There are 3 distinct features of our test corpus. Firstly, our test corpus consists largely of papers from the field of Zoology. Secondly, the test-corpus is not limited to English Language, containing papers written in many languages. And lastly, there are papers ranging from 18th century to the present in our corpus which papers in different formats and following different conventions. Although, our test corpus is mainly



**Fig. 1.** Process Outline

restricted to papers related to Zoology and various ant species the algorithm described in this paper can handle papers presented on other subjects also.

The basic strategy followed in converting the input files are as follows: At first, the documents are hand scanned using Optical Character Recognition (OCR) and converted into a PDF Format. The PDF documents are then converted into html using Abby PDF Reader which is propriety software for converting PDFs. Since the text of the PDFs are produced by OCR, there can be errors due to scanning in the test corpus. The formatting of the text is usually not consistent. Also, the capitalization of the text cannot be relied upon as useful indicator for parsing. Certain spelling mistakes are also encountered whenever the print is unclear. After this, the GoldenGate Editor [11] is used to add annotations and then export to an internal format which is converted to TaxonX [5][6] by a purpose built XSLT transformation which is being used internally in Plazi. The TaxonX Schema based xml files are then used by the Reference Block Identifier as an input.

## 4.2 Reference Block Identification

The document is then processed to identify reference block. A reference block can be defined as a set of paragraphs in the document contains all the references to other papers. Usually the reference block is present in the end of a research paper and is often preceded by a keyword: "References". This is often followed by references preceded by index such as 1,2,3.... ; [1], [2], ...;A, B, .... and so on.

It has been observed that every reference block starts with a verbal cue which we will refer to as a keyword. The keywords differ with various languages and publications such as "References", "Works Cited" etc. Thus, a very simple approach for the reference extraction, (also employed by the ParaCite Project) is to look for Keywords such as "References" in the research paper and then taking the reference block to be as the block from the start to the end of the paper. This approach although

correct in essentials often misses out on certain Reference blocks which are preceded by keywords not present in the database like: "Bibliography", "Works Cited", "Bibliographia" etc. Thus, a minor improvement to the approach will be to scan for all such keywords to identify the start of a reference block. Thus, an editable list of keywords should be maintained which helps to identify the start of a reference block. But with reliance on keyword alone will mean that there will always be a chance that the program might encounter a keyword not present in our database.

Also, a document may only have different keyword to denote the start of a reference block; it will not necessarily contain only references until the end of the paper. For instance, a document can contain a list of tables, a list of figures or appendix information after the reference block. Also, a document such as a journal or a conference proceeding often contains more than one paper and hence, more than one reference blocks in it. Thus, our second assumption that a reference block once started continues till the end of the document is also not correct every-time. A minor improvement to the above mentioned algorithm will be to scan for words such as "Tables", "Figures", and "Appendix" and stop the reference block when they are encountered. Since, we cannot rely on formatting information we cannot provide the start of another section by checking the difference in the format of the words. This modification too, will not give us very satisfactory results and will lead to errors.

Thus, a probability model based approach in which we look at each paragraph as a potential candidate for being a reference may be preferable. Every paragraph which is a reference will contain a year to denote the year of publication. By manual inspection, it has been found, that every reference contains a year and thus the absence of a year from a paragraph positively means that the paragraph is not a reference. But the inverse is not always true, since a general paragraph might also contain a reference to a year, or might even cite a reference in the format such as: (Allen, 1987). Thus, the presence of the year in the normal paragraph (i.e. a paragraph which is not a reference) will either be in a sentence or in the form of a citation. Thus, combining this information along with the locality of a reference block i.e. the references being present one after another, and being preceded by a keyword, we can make a good probability model. The other criteria we put into the model are the conformance of a paragraph to a reference structure. Usually, by reference structure, it is meant that a paragraph denoting a reference is small in length and it follows a certain template.

If we classify the parameters as:

P1: Absence of year

P2: Presence of keyword

P3: Previous words of a year conforming to Reference Structure

P4: Correlation in the length of paragraph with Average Reference Length

P5: Number of reference paragraphs directly above the candidate paragraph

P6: Number of reference paragraphs directly below the candidate paragraph

P7: Correlation in Reference Template with the candidate paragraph

And we classify the weights of the parameters as: W1, W2, W3, W4, W5, W6, W7 correspondingly.

The probability of a candidate paragraph to be a reference can then be defined as:

$$P(\text{Candidate}) = \sum W_i P_i / \sum W_i$$

If  $P(\text{Candidate}) \geq \text{Threshold Value}$ , then Candidate paragraph is a reference.

Thus, the first step is to search for a year which is a four digit number from 1800 to the current year. Apart from this, we make use of other parameters such as presence of a keyword before the paragraph, references immediately before and after the paragraph being considered, length of the paragraph and the perceived author name from the paragraph. All this information has been put in a probability model which is customized to get the overall probability of a particular paragraph to be a potential reference. If the probability is above a certain threshold(0.7 in our case), we assume it to be a reference. All the references thus extracted are cleaned up to remove the extraneous formatting and redundant information. The cleaned up references are put into a temp file for internal use by the next module.

### 4.3 Reference Parsing

As discussed earlier, the presence of abbreviations, inconsistent formatting and semantically overloaded punctuation and separators have presented difficulties. Thus, we have followed an integrated approach towards this problem. The approach consists of:

1. Template matching i.e. use of regular expressions to classify various portions of the text as particular fields.
2. Use of domain based information like a list of publications to classify a portion as a Publication in case of failure by the first approach.

The parsing for a reference starts simultaneously from left and right. The parsing is started from the left to obtain the Author Name and Year information. It has been observed from the test corpus that every reference has author name(s) being immediately followed by the year of publication. Often, there are more than one Authors followed by a year. A year is a four digit number which will have the value within the range of normal year values. A year can also be followed by a special character as 1996a or 1996b. A reference might also contain more than one year in a single line such as 1995, 1996 to denote two papers published by the same authors. Keeping all this in mind, the year and author names are extracted from the reference. Simultaneously, parsing is started from the right to look for page-numbers and related cues. For instance, a page number is represented as pp. p. or without any prefix, simply as two numbers separated by a hyphen. Regular expressions are used thus to find out Author Names, Year of Publication, Volume and Page numbers with the reference being parsed both from the front and the back. The parsed information is then removed from the reference paragraph.

For the other parts of the reference, the approach is not so simple. The use of separators often cannot be used as a reliable measure for distinguishing between Title and Publication. This is because no single separator is used consistently to distinguish a title from a publication. In many cases, a dot (.) is used both as a separator and also as an abbreviation marker. Similarly, common separators like comma (,), hyphen (-) and colon (:) are all semantically overloaded serving both as a field separator as well as working as punctuations. It has also been observed that many references have all the words of a title capitalized, while some have every character capitalized and the rest have every character in small-case letters. Since, capitalization is neither preserved during scanning and nor is a uniform means of distinguishing, it cannot be used for the

parsing. Thus, capitalization, formatting or punctuations cannot be used as a reliable means for parsing.

A combination of domain specific knowledge and a prediction model based on the length of the typical fields has been followed with automatic detection of certain false results. For this, a huge database of existing Zoological Publications has been formed largely from Zootaxa and the Natural History Museum Database to find out the publications present in a reference. We use a word-to-word based matching algorithm which matches the publication with the most eligible candidate i.e. the publication with most matched words. It has been noted that some of the words are redundant such as various prepositions and can generate false matches. Thus, those words are not taken into account. When we have a single word match, we largely ignore it, but we take it into account only when the publication has a single word in it. Note that a match is only considered for consecutive words, i.e. there can be no gap between two matched words except that of whitespace or punctuations.

If the publication finds no match, then we take the prediction model into account. The prediction model is based on the average publication length, their distance from the start of the reference and the major separators used. The statistics of the correctly parsed references are logged in while parsing. These are then used for the references that fail. Sometimes, a reference might not have any publication listed; in such cases also, we need to correctly parse the titles.

Thus, with the help of domain knowledge, regular expressions and prediction models, we achieve good accuracy in parsing.

#### **4.4 Citation Parsing**

It has been observed that a citation always consists of an author name followed by a year. The indexed citations are easy to associate with a corresponding reference. A textual citation, however requires more work. It can be found in a paragraph by looking for years in the paragraph. After finding a year, we need to find out whether it is preceded by an author name or not. It has been observed that there are certain temporal prepositions which commonly appear before years such as “in, since, during, until and before”. If these temporal prepositions come in front of a year, it automatically means that this phrase is not a citation.

Identification of author names is done by matching of Authors field from parsed references. Every citation corresponds to a certain reference and thus the localized information obtained by parsing of the Reference section can be used to identify the Author. If the Author match is found, the corresponding match is taken as a citation, otherwise, the phrase and the year succeeding it is not considered as a citation. The punctuations are not at all considered in this process. Thus, this algorithm is able to identify both the textual as well as parenthetical citations in the paper. At the end of this process, an output file is generated which contains all the parsed information consisting of all the references and the corresponding citations in the research paper.

## **5 Experiment and Results**

Our test corpus consists of papers for which a TaxonX schema based document has already been generated using the above-mentioned approach. The corpus consists of

papers written in various different languages and are encoded in UTF-8 encoding. The corpus has been classified according to the languages as in Figure 2. The algorithm provides an accuracy which is considerably better than the accuracy provided by the various approaches used individually.

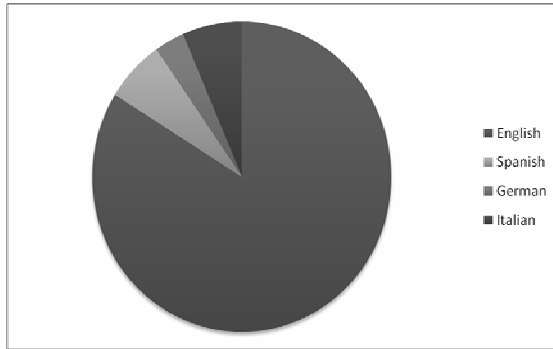


Fig. 2. Languages in Corpus

Table 1. Reference Block Identification

Title	ParaCite Unmodified	Modified Code I	Modified Code II	Our Approach
Number of References(Hand Counted)	664	664	664	664
True Positives Identified	330	660	660	660
False Positives Identified	310	550	86	3
Percentage of False Positives	48.4%	45.5%	13.0%	0.5%
Percentage of False Negatives	50.4%	0.6%	0.6%	0.6%

Table 1 presents the number of references detected by various programs from a research paper in our test corpus. The unmodified ParaCite code only identified 330 of 660 references of the total references present in the document. Thus, the original ParaCite code has low precision and low recall. Our first modification improved the recall, but not the precision. Our second modification improved the recall to 100%, the same as our own, but with 10 false positives compared to our 3. ParaCite suffered its low recall because it identifies references by looking only specific keywords like "References" or "Works Cited". Our first modification enhanced knowledge base of reference block candidates to include all possible keywords. Our second set of modifications reduces false positives by excluding several common types of document segments that commonly follow a reference block, but which do not

contain references. Such sections include a list of figures and tables or appendices. Our own method adopts some of these enhancements, along with altogether different probability based-model, which we feel is more widely applicable. See section 3.2.

**Table 2.** Reference Parsing

<b>Fields</b>		<b>Number Correctly Identified</b>	<b>Percentage Correctly Identified</b>
<b>Authors</b>		660	99.5%
<b>Year of Publication</b>		660	99.5%
<b>Title</b>		504	76.0%
<b>Publication</b>	<b>Combined Approach</b>	490	73.9%
	<b>Regular Expression Heuristic</b>	445	91.8%
	<b>Knowledge Based System</b>	45	9.2%

The Table 2 represents the number of fields correctly parsed in the references parsed by the algorithm. The various approaches followed while parsing Publications are shown in the columns Domain Based and Prediction Based. The domain-based knowledge obtained from the database of reputed publications for taxonomic papers was used to identify the Publisher/journal effectively.

Apart from this, in case no such match was found, prediction was made using the localized knowledge such as the average length, starting point of publications, separator symbols used etc. and the algorithm tried to predict the publication using it. It was often found that the prediction based knowledge supplemented the first technique to give us better results. It should be noted however, that the existing system will need the domain-based knowledge for this technique to work. Thus, a corpus must always be supplemented with this knowledge, otherwise, the results might not be as good as shown above.

While the identification of “Title” and “Publication” has somewhat lower accuracy, “Author” and “Year of Publication” show high accuracy results. This is because many of the references in our corpus are micro-citations and have semantic overloading of information in the punctuation marks like dot(.). An example of the same will be:

Simon, E. 1891. On the spiders of the island of St. Vincent. Pt. 1 Proc. Zool. Soc. of London, Nov. 17, 1891: 549 - 575

In this, the title ends at St. Vincent. But the title itself contains dot as an abbreviation specifier and also as a separator between Title and Publication. We encounter many such cases of micro-citations in our test-corpus.

## 6 Conclusion and Future Scope

This paper studies the limitations of existing open-source solutions such as Paracite in bibliographic reference parsing along with the description and implementation of a new approach. This approach combines multiple techniques to enhance the accuracy and provide better recognition rates. More work needs to be done to achieve 100% accuracy results. The future scope includes the integration of a more efficient self-learning model into this approach to make the program learn from its mistakes. The refinement of the project can further be done by using a bigger test corpus. Also, the requirement of an existing knowledge domain can be lessened in the future by training it with multiple domains of human knowledge.

## References

1. Jewel, M.: Paracite (2003), <http://paracite.eprints.org/developers>
2. Giuffrida, G., Shek, E.C., Yang, J.: Knowledge-based metadata extraction from PostScript files. In: DL 2000: Proceedings of the fifth ACM conference on Digital libraries, pp. 77–84. ACM Press, New York (2000)
3. Powley, B., Dale, R.: Evidence-based information extraction for high accuracy citation and author name identification. In: Proceedings of RIAO 2007: The 8th Conference on Large-Scale Semantic Access to Content, Pittsburgh, Pa., USA (2007)
4. Sautter, G., Böhm, K., Agosti, D.: A combining approach to find all taxon names (FAT). *Biodiv. Inf.* 3, 46–58 (2006)
5. Sautter, G., Böhm, K., Agosti, D.: A Quantitative Comparison of XML Schemas for Taxonomic. *Biodiversity Informatics* (2007)
6. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: *Knowledge Discovery and Data Mining*, pp. 169–178 (2000)
7. Hetzner, E.: A simple method for citation metadata extraction using hidden markov models. In: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital Libraries (2008)
8. Takasu: Bibliographic Attribute Extraction from Erroneous References Based on a Statistical Model. In: Proceedings of Joint Conference on Digital Libraries (2003)
9. Huang, I.A., Jan-Ming, H., Kao, H.Y., Lin, S.: Extracting citation metadata from online publication lists using BLAST. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS, vol. 3056, pp. 539–548. Springer, Heidelberg (2004)
10. Matt, E.D., Winkels, R., Van Engers, T.: Automated Detection of Reference Structures in Law. In: Proceedings of the Conference at University Pantheon, Assas, Paris II France, pp. 41–50 (2006)
11. Sautter, G., Agosti, D., Böhm, K.: Semi-Automated XML Markup of Biosystematics Legacy Literature with the GoldenGATE Editor. In: Proceedings of PSB, Wailea, HI USA (2007)