ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

Implementation of Citation Parser on the Basis of Knowledge Base Hierarchy

Anand V. Saurkar[1], Prof. A.R. Itkikar
Department of Computer Science & Engineering
Sipna's COET, SGBAU, Amravati (MH), India

***Abstract*-- Use of the bibliographical information of publications available on the Internet is an important task in academic research. Accurate reference metadata extraction for publications is essential for the integration of information from heterogeneous reference sources. It is an essential task in the research paper development to point out references for proper document. Giving a proper acknowledgement to a document or part of document is called citation. A "citation" is the way to tell readers the source from which certain material has come. In general citation implies a relation between part or whole of the cited document and a part or whole of the citing document. To cite a particular document there are several methods. We shall develop a citation tool based on knowledge based citation parsing technique. By developing a current system we shall extract the cited document which is useful for scientific publication or document citing.**

***Keyword*:- Citation, Parser, Text search, knowledge Base .**

I. INTRODUCTION

In the publication or academic communities, citations are a key part of linking distinct pieces of knowledge into a well-structured record of a field. Although citations can come in many forms, a common standard places a reference to each paper, article or book used by the authors, in a separate section of the work [1]. This list, often known as a bibliography, acts as an acknowledgement to these materials and provides exact publication information to identify each source and allow a reader to locate it for further study.

Users often use citations to find information of interest in Digital Libraries, while researchers depend on citations to determine the impact of a particular article. Parsing citations is essential for integrating bibliographical information published on the Internet.

Parsing citations is essential for integrating bibliographical information published on the Internet. Most citation management techniques are based on the assumption that we can correctly identify the main components of a citation, such as authors' names, title, publication venue, date, and the number of pages. However, for a variety of reasons, it is difficult to design a parser that can automatically parse citations scattered over the Internet [1]. Potential problems include data entry errors, diverse citation formats, the lack of (enforcement of) a standard, imperfect citation gathering software, common author names, abbreviations of publication venues, and large-scale citation data. We propose a knowledge-based citation parser, to extract components of citations in any given formats. The basic idea of this citation parser is to capture the structural properties from semistructured format and transform these

properties into a sequence template. The structural properties of a citation string include the order of punctuation marks and local structure in each field of a citation string. We use an encoding table and reserved words, which is automatically trained from the data set, to represent each semantic unit as a unique symbol; and use a blocking process to capture local structure in each citation field.

There are various methods like machine learning technique and knowledge based technique to cite a document. We shall propose a system which is based on a knowledge based hierarchy technique.

II. RELATED WORKS

Citation indexing can improve scientific communication by revealing relationships between articles, drawing attention to important corrections or retractions of published work, identifying significant improvements or criticisms of earlier work, and helping limit the wasteful duplication of prior research.

The problem of citation parsing has been the focus of past research initiatives, as documented in the literature [2]. Existing citation parsers can be generally divided into two categories: template matching and machine learning based approaches.

A. TEMPLATE MATCHING APPROACH

A template matching approach takes an input citation and matches its syntactic pattern against known templates. The template with the best fit to the input is then used to label the citation's tokens as fields. The canonical example of a template based approach is ParaTools. Disadvantage of ParaTool is on the basis of available template it generate result[1].

This technique works fairly well for citations which adhere to simple citation patterns, but is susceptible to errors when it tries to extract fields from citations with many punctuations, since there may be multiple templates that fit equally well. If the wrong template is chosen, entire fields will be tagged incorrectly.

B. MACHINE LEARNING BASED CITATION PARSING TECHNIQUES

The limitations of the template-based approach have encouraged researchers to try alternative models for citation parsing.

Hidden Markov Models (HMMs) are a powerful probabilistic tool and has been applied extensively on various language related tasks [2] [4]. HMMs are a finite state automaton with stochastic state transitions and symbol emissions. HMMs may be used for citation parsing by formulating a model in the following way: each state is associated with a citation field name (hereby called "tag") such as title, author or date. A labeled training dataset is first used to train a HMM. This model is then used to recover the most-likely state sequence that produces the sequence of observation symbols. the HMM method is less effective due to its assumptions of independent and non-overlapping features[6].

Conditional Random Fields (CRF) model to parse citations in another experiment. CRFs are undirected graphical models trained to maximize a conditional probability and have been applied on tasks such as name entity extraction. Training time is a concern, as CRFs converge slowly. It requires approximately 500 iterations for the model based on the same training set to stabilize [5].

The Maximum Entropy Model provides flexibility given sufficient training datasets and

serves as a balance between the two mentioned machine learning models. Since then, maximum entropy techniques have widely used for natural language tasks such as identifying sentence boundaries and text classification. Disadvantage of this system is , additional supporting databases such as a journal name database, publisher database, country name database, etc. can also be collected to help the system identify and extract fields[2].

III. ANALYSIS OF PROBLEM.

In previous work, techniques like Template Based Citation Parser, achieved approximately 80% of parsing precision. But it has a drawbacks.

First, the template construction in our previous work relies on an author name database to identify possible author names. So the quality of the database greatly affects the parsing accuracy, and author name database of high quality is never easy to obtain in practice.

Similarly, a Machine learning base citation technique has following drawbacks.

First, several heuristic rules are applied in our previous work to transform training citation strings into related templates, but these rules only work for several special cases. It causes problems when extracting metadata from a citation string that does not follow those special cases.

Second, during the matching process in previous work, there has high probability to mismatch a wrong template to a citation string because there are several templates having the same similarity score, and no other information could be used to distinguish them. We call this the "template conflict" problem. Generally, the larger the template database, the more serious the problem is.

IV. PROPOSED WORK AND OBJECTIVES

To solve above problem regarding automatic citation extraction we propose a system which is based on knowledge based hierarchy technique. A feature of this system is its capability to represent and match complicated structures, such as hierarchical matching, regular expressions, semantic matching. On the successful development of this proposed system, we shall be able to extract author, title, journal, volume, number (issue), year, and page information from different kinds of reference. A hierarchical approach is used to detect source of information from source data base. Knowledge data base is a combination of all data available from a particular domain set.

Objective of this proposed system is collection of reference data, saving it and applying parsing on that data to cite a particular inputted string.

A CITATION PARSER

Citation plays an important role in scientific publication. Knowledge-based approaches utilize domain knowledge to derive ontology that describes the data of interest, where the knowledge includes relationships, lexical appearances, and context keywords. By parsing the ontology, several rules and an extractor can be generated, which are then used to perform information extraction.

Our proposed system work as follows:

1. Categorization of paper
2. Text search method
3. To find number of words matched in the submitted document

4. To detect the number of sentences matched in the submitted document.

CATEGORIZATION OF PAPERS

When a paper is submitted to the system, cross checking the paper with all the registered papers can become redundant. Each paper belongs to a field of research work. The registered papers can be divided into a finite number of groups so that the processing load on the server can be reduced.

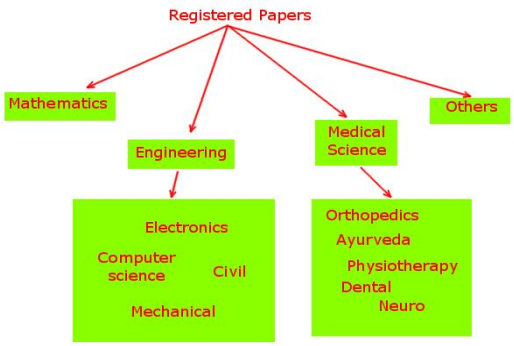

Fig 1: Categorization of Papers

TEXT SEARCHING METHODS

Parsing the text in a document file is a crucial task. In order to compare every word or sentence, we should be able to find it in the source file easily. The result for the search can be either “found” or “not found”. But accessing the source file word by word and also frequently can take the complexity of such a parsing to O(nn) which is an exponential growth. Such a parsing can take huge amount of time to give an output.

A theoretical algorithm for searching an element which takes O(1) complexity, can get output in a single instruction, but it is not possible to implement in any of the language because it is an NP-HARD problem in Computer Science. Still research is carried in this field and the problem is stuck on whether P = NP or not. The minimum worst-case complexity in searching an element from a given set of elements is O(log2n) which can be applied in a programming language and that is achieved by Binary Search technique. Binary Search Technique follows Divide and Conquer policy to search an element in a given set of object. The constraint in this Binary Search Technique is that the set of objects that it has to be implemented on should be already sorted.

Now here the problem arises on sorting the words in the source document. Like searching, sorting also has a theoretical algorithm which takes O(1) complexity. But the algorithm fails while implementing it in any language due to the class (NP-HARD) in which it has been classified. Practically to sort a set of elements, few algorithms in their worst-case takes a minimum complexity of O(nlog2n). Some of the sorting techniques with a complexity of O(nlog2n) are Merge Sort, Heap Sort, Intro Sort, Time Sort, Binary Tree sort, Patience Sort, Smooth Sort and Tournament Sort. Here in the Plagiarism Detection System we will make use of Merge Sort. The reason for using Merge Sort over other sorting algorithm is:

- No matter what the input size is, the complexity of Merge Sort will remain O(nlog2n)
- Merge Sort is stable

In Text Searching methods we are also going to make use of Regular Expressions. Regular Expression is a great tool to search text in the registered document. An algorithm will be run over each and every sentence in the submitted document. This algorithm will dynamically generate a regular expression for every sentence in the submitted document.

TO FIND NUMBER OF WORDS MATCHED IN THE SUBMITTED DOCUMENT

Manipulation of words from a registered paper is one of the plagiarism techniques. Changing the voice (active to passive and vice versa), manipulating the words in the sentences, adding words to the sentences to make it look different than the original one, etc are the techniques used by the culprit to make the paper look like an original work which is not acceptable. In order to remove these anomalies we have build this module. Text searching technique mentioned above comes handy in this module as well as in the upcoming modules.

Let us assume that we have 'X' number of registered documents say {D1,D2,D3,…..,Dx}, in our database which are free from plagiarism and each document contains 'n' number of words. Thus total number of words can be given as 'Xn'. This shows us that we need to sort 'Xn' number of words for every document that will be submitted to the server. By using the Merge sort we can get the least complexity of

$$\sum_{0}^{X-1} \Theta(n.\log(n)) \approx \Theta\,(xn.\log 2(xn))$$

This complexity can be further reduced by selective approach. The registered papers can be divided into finite number of groups and accordingly the submitted text can be checked against the papers of the group which it belongs to.

This module has to return the amount of words matched from a registered paper. These words can either be deliberately added in the submitted document (which can be a case of paraphrasing) or unintentionally. But a certain upper bound must be set in order to maintain the proficiency of the respective topic of research. In this module a logarithmic value and a percentage of the number of words that have been matched are taken into consideration. This is because the number of words in the submitted document can be huge and taking a direct value cannot be appropriate. Thus percentage to the words copied or a logarithmic value of the words that are matched comes handy and also it is easy to understand.

TO DETECT THE NUMBER OF SENTENCES MATCHED IN THE SUBMITTED DOCUMENT.

A regular expression is a set of pattern matching rules encoded in a string according to certain syntax rules. For every sentence in the submitted document, we need to build a regular expression dynamically in order to match it with the registered documents. The algorithm for building dynamic regular expression string is given as follows

algo:(String) PatternCreate(String x) // Returns a String and takes String x as an argument

{

string string_to_return; //initialize a string which has to be return

while(next word in x does not ends with '.') //Parse the string from left to right until

{

// a period is encountered

read next word --> z;

char firstChar = z[0]; //take the first alphabet from the next word

append "[firstChar]\\w+" to string_to_return; //add first character to the string

if(z does not ends with '.') //if not the last word

{

append "\\s" to string_to_return; //append space

}

}

return string_to_return;

}

Algorithm to create Regular Expression pattern.

Let us assume that submitted document contains ‘n’ sentences and every sentence contains ‘m’ number of words. Thus we need to create ‘n’ number of regular expressions and these regular expressions will be matched with each of the registered document in the database. The complexity of the algorithm mentioned above will be O(m). Overall there will be ‘nm’ number of words in the submitted document. Thus to create a set of regular expression for the submitted document will be O(nm). Finally we need to match this set of regular expression with each document in the database.

SYSTEM WORK FLOW DIAGRAM:

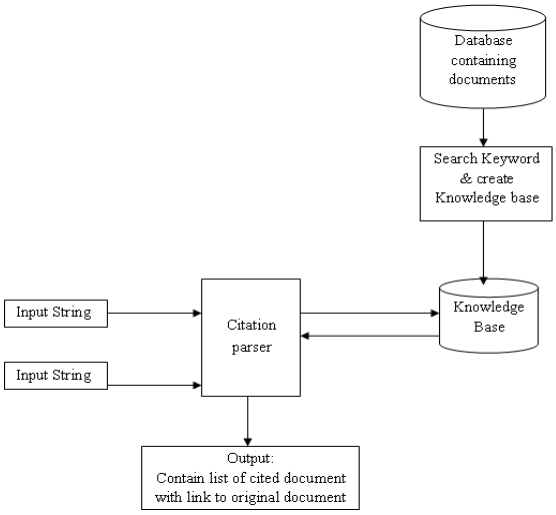

Fig1: Working flow of citation parser

According to a working flow diagram of propose parser, a knowledge base is created from database. For creation of knowledge base we use keywords from documents. More key words are form from title of paper and abstract. As documents are loaded in database, system analyze domain of the paper.

System has a search field which is work as search window. String or numbers of string are writing in search window. Citation parser parses a string and match with knowledge base. If input strings are match with data in knowledge base then gives a result in form of list of match documents and also provide a link to the original document.

CONCLUSION

Parsing citations is challenging due to the diverse nature of citation formats. In this report, we present an implementation of citation parser on basis of knowledge based hierarchy. The basic concept of this parser is to transform semistructured properties of a citation string into a

sequence template, and apply parsing technique to further resolve the structured information.

REFERENCES:

[1] BibPro: A Citation Parser Based on Sequence Alignment Chien-Chih Chen, Kai-Hsiang Yang, Chuen-Liang Chen, and Jan-Ming Ho. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 24, No. 2, February 2012.

[2] Citation Parsing Using Maximum Entropy and Repairs ,Ng Yong Kiat , Department of Computer Science ,School of Computing , National University of Singapore .2004/2005.

[3] A Knowledge-based Approach to Citation Extraction, Min-Yuh Day1,2, Tzong-Han Tsai1,3, Cheng-Lung Sung1, Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, shwu@cyut.edu.tw.

[4] E. Hetzner, "A Simple Method for Citation Metadata Extraction Using Hidden Markov Models," Proc. Eighth ACM/IEEE-CS Joint Conf. Digital Libraries, 2008.

[5] F. Peng and A. McCallum, "Accurate Information Extraction from Research Papers Using Conditional Random Fields," Proc. Human Language Technology Conf. and North Am. Chapter of the Assoc. for Computational Linguistics (HLT-NAACL), pp. 329-336, 2004.

[6] P. Yin, M. Zhang, Z. Deng, and D. Yang. Metadata extraction from bibliographies using bigram HMM. In Proc. of the 7th Intl. Conf. on Asian Digital Libraries, LCNS 3334, pages 310-319, 2004.

[7] Citation Matching in Sanskrit Corpora Using Local Alignment Abhinandan S. Prasad and Shrisha Rao International Institute of Information Technology, Bangalore abhinandan.sp@iiitb.net, srao@iiitb.ac.in

[8] A New Approach towards Bibliographic Reference Identification, Parsing and Inline Citation Matching, Deepank Gupta, Bob Morris, Terry Catapano, and Guido Sautte,r Netaji Subhas Institute of Technology, Plazi.

[9] V. Borkar, K. Deshmukh, and S. Sarawagi, "Automatic Segmentation of Text into Structured Records," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2001.