ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Extracting Algorithms by Indexing and Mining Large Data Sets

Vinod Jadhav[1], Dr.Rekha Rathore[2]

P.G. Student, Department of Computer Engineering, RKDF SOE Indore, University of RGPV, Bhopal, India

Associate Professor, Department of Computer Engineering, RKDF SOE Indore, University of RGPV, Bhopal, India

ABSTRACT: Efficient algorithms are extremely important and crucial for certain software projects. There are many Search Engines (SEs) are available but it is still hard to locate and concentrate only on materials relevant to a specific task. Recently, AlgorithmSeer, a search engine for algorithms, has been investigated as part of CiteSeerX with the purpose of providing a large database of algorithms. It has ability to automatically find and extract these algorithms in this increasingly large collection of scholarly digital documents would enable algorithm indexing, searching, discovery, and analysis. The System proposes a novel set of scalable techniques used by AlgorithmSeer to identify and extract algorithm representations in a differerent pool of scholarly documents. Specifically, hybrid machine learning approaches are proposed to discover algorithm representations. Then, techniques for extracting textual metadata for each algorithm are discussed. And Finally, a different version of AlgorithmSeer that is built on Solr/Lucene open source indexing and search system is presented.

KEYWORDS: Algorithm Seer, Algorithm Procedures (Aps), Pseudo Codes(PCs),Machine Learning.

1. INTRODUCTION

Algorithms plays an Vital role in many software projects. For example, an efficient ranking algorithm helps to improve the performance of software used for indexing and searching billions of documents. Hence, it is important for software developers to keep abreast of latest algorithmic developments related to their projects[5]. There has been much work in past to help software developers search. for existing source code, however to the best of our knowledge, no effort has been made to develop tools and techniques that can help software developers search for literature about the algorithms used in source code or in documents that describe new algorithmic developments.

The relevant algorithm procedures and pseudo codes are not easily available with their analysis and it requires more efforts and time to search them. Number of algorithm procedures and pseudo codes are being published every year in national and international journals. So searching for efficient and relevant algorithm procedures and pseudo code become difficult as efforts are needed to compare and analyze them to determine the most efficient one. So, each and every research papers have to be referred. Yet the probability of acquiring relevant andefficient algorithm procedures and pseudo codes is less. To address these issues, appropriate mining techniques have to be applied. These techniques involve knowledge discovery from web and available research papers from national and international journals in order to obtain most suitable match for the input request from user. To achieve this, input request is accepted, indexing is done using proper mechanisms and most relevant algorithm procedures and pseudo codes are listed. Also,user is provided with downloading facility for algorithm procedures and pseudo codes. Hence, algorithm procedures and pseudo code extraction and analysis become an important part of this implementation.

II. BACKGROUND AND RELATED WORKS

Number of algorithm procedures and pseudo codes are being published every year in national and international journals[1]. So searching for efficient and relevant algorithm procedures and pseudo code become difficult as efforts are needed to compare and analyze them to determine the most efficient one.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

2.1 Element Extraction in Scholarly Documents

Extractionby identifying informative entities such as mathematical expressions,, tables , figures, and tables of contents from documents have been studied.. They also proposed a method to index and search for the information, which is extracted .Sojka and Liska proposed Math Indexer and prescribed. The user may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and Searcher that interprets and collects mathematical expressions.

2.2 Search Systems for Scholarly Information

CiteSeer is a not only digital library but also search engine for all academic documents. Traditionally, the focus of CiteSeer_ has been on computer science, information science and related disciplines; however, in recent years there has been an expansion to include other academic fields also. The key features of this algorithm_are that it automatically performs information extraction on all documents added to the collection and automatic indexing, which allows for citations and other information among papers to be linked[1].

Fig. 1. Sample search results for the query “shortest path finding algorithm” on CiteseerX.

Though useful algorithms could be found in many search engines (e.g., CiteseerX9, Google Scholar10) and algorithm-related discussion forums (e.g., Stack Overflow11, Quora12), the search results can be contaminated with irrelevant items.

Fig. 1 illustrates an example of a search session (top five results) for shortest path finding algorithms using the query “shortest path searching algorithm” on CiteseerX search engine that hosts 5 million academic publications whose major proportion is computer science and related disciplines[4]. According to the figure, only two out of five top results are relevant (actual papers that describe shortest path algorithms), while the other three results are simply documents containing those search terms. These search engines would return the whole documents (papers or websites) as results, requiring the users to invest additional, unnecessary effort to read the entire documents to find the desired algorithms. This system is the first to explore the possibility of building a search engine specifically for algorithms extracted from scholarly digital documents.

III. THE PROPOSED SYSTEM

In this system, a prototype of an algorithm search engine, AlgorithmSeer, is presented. Fig. 2 explains the high level of the proposed system. The proposed system analyzes a document to identify any algorithms that may be present

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

in the document. If an algorithm is found in a document, the document text is further analyzed to extract additional information about the algorithm.

All the algorithms thus found with their associated metadata are indexed and made available for searching through a text query interface. For a given user query, the system utilizes evidence from multiple sources to assign relevance scores to algorithms and results are presented to the user in decreasing order of relevance.

First, scholarly documents are processed to identify algorithm representations . Then, the textual metadata that provides relevant information about each detected algorithm representation is extracted . The extracted textual metadata is then indexed, and made searchable.

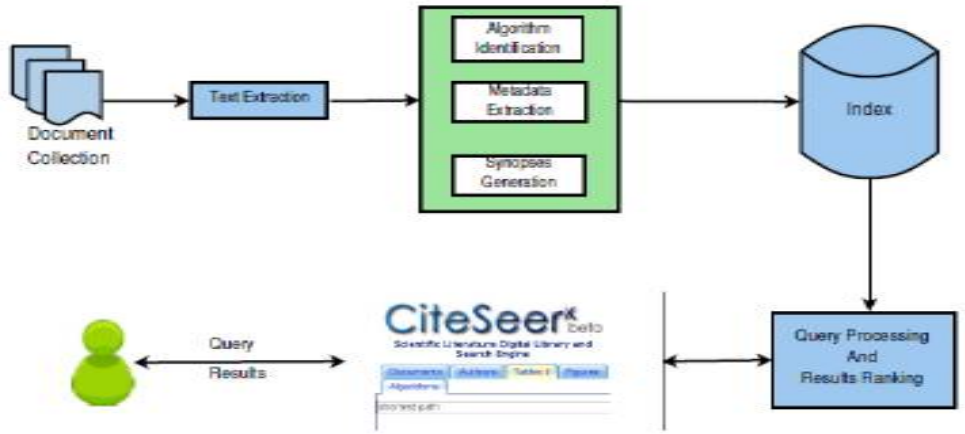


Fig. 2. Architecture of the proposed system.

Fig. 2. Architecture of the proposed system.

IV. ALGORITHM IDENTIFICATION IN SCHOLARLY

This section discusses the methods for automatic discovery of PCs and APs in scholarly documents. The system specifically handles PDF documents since a majority of articles in modern digital libraries including CiteseerX are in PDF format. First, plain text is extracted from the PDF file. Inspired by Hassan , we use PDFBox13 to extract text and modify the package to also extract object information such as font and location information from a PDF document. Then, three sub-processes operate in parallel, including document segmentation, PC detection, and AP detection.

The document segmentation module identifies sections in the document. The PC detection module detects PCs in the parsed text file. The AP detector first cleans extracted text and repairs broken sentences (since the method assumes that a document is represented with a sequence of sentences), then identifies APs. After PCs and APs are identified, the final step involves linking these algorithm representations referring the same algorithms together. The final output would then be a set of unique algorithms[5].

4.1 Detecting Pseudo-Codes (PCs)

Since PCs can appear anywhere in a document, these identifiers usually serve the purpose of being anchors which can be referred to by context in the running text. Here, three approaches for detecting PCs in scholarly documents are presented: a rule based method (PC-RB), an ensemble machine learning based method (PC-ML), and a combined method (PC-CB)[4]. Note that though OCR based techniques (where each page of the document is first converted into an image, where image processing based techniques are used to locate the boundary of the PCs, then an

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

OCR technique is used to extract the content within each candidate region) have been explored, textual content can be directly and quite accurately extracted from most PDF files.

4.1.1 Rule Based Method (PC-RB)

Recently, Bhatia et al. proposed a rule based PC detection approach, which utilizes a grammar for document-element captions to detect the presence of PC captions. The methods are reffered as our baseline (PC-BL) for the PC detection task.

4.1.2 Machine Learning Based Method (PC ML)

The PC-RB contains a high precision, however it still suffers from a low coverage resulting in a poor recall. It is found that 25.8 percent of PCs in dataset do not have accompanied captions. The PC-ML first detect and extracts these sparse boxes, then classifies each box whether it is a PC box or not. A PC box is a sparse box that contains at least 80 percent content of a PC. The following sections explain how sparse boxes are identified, the feature sets, and the classification models. The output of the PC-ML is a tuples of start; ending line numbers of the detected PC.

4.1.3 Combined Method (PC-CB)

Though the PC-ML method can capture PCs even though they do not have accompanied captions, some PCs are not first captured in a sparse box would still remain undetected. such PCs cannot be captured using the sparse box extraction. However, 27 (out of 35) of these undetected PCs have accompanied captions and hence might still be detected using the PC-RB method. The proposed method combines method (PC-CB) of the PC-RB and the PC- using a simple heuristic as follows:

STEP1 For a given document, run both PC-RB and PC-ML methods.

STEP2 For each PC box detected by PC-ML, if a PC caption detected by PC-RB is in proximity, then the PC box and the caption should combined.

4.2 Detecting Algorithmic Procedures (APs)

Two methods are developed for detecting AP indication sentences: a rule based method (AP-RB) and a machine learning based method (AP-ML). Both methods rely on the sentences which correctly extracted from the document. However, most scholarly documents are multi-columned and contain document elements such as tables and diagrams[6].

Hence, before extracting sentences, garbage text fragments are removed from the document. The heuristic is also developd that stitches up an incomplete sentence which are broken into multiple lines.

4.2.1 Rule Based Method (AP-RB)

Often, AP indication sentences exhibit certain common properties:

- The sentences usually end with follows:, steps:, algorithm:,follows:, following:, follows., steps:, below:.
- The sentences usually contain at least an algorithm keyword.

We create a set of regular expressions to capture sentences according to the rules above.

4.2.2 Machine Learning Based Method (AP-ML)

Some AP indication sentences do not conform to the rules described in Section 4.2.1. Therefore, an alternative approach based on machine learning is proposed to directly learn to capture the characteristics of APs.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

4.3 Linking Algorithm Representations

A simple heuristic is implemented to link algorithm representations referring to the same algorithm together. Specifically, two algorithm representations are linked if:

1) They represent the same algorithm. For Ex. an AP may be used to provide more detail to a PC.

2) They are part of the same algorithm. For Ex. an algorithm may be broken into sub-parts, each is represented using a different algorithm representation.

4.3.1 Identifying Sections in Scholarly Documents

A machine learning based approach is used to identify section boundaries by detecting section headers. The algorithm was first proposed by Tuarobet almost scholarly documents have following common properties:

1) Each section has a section header and section number.

2) Section headers have distinct font styles from the surrounding content.

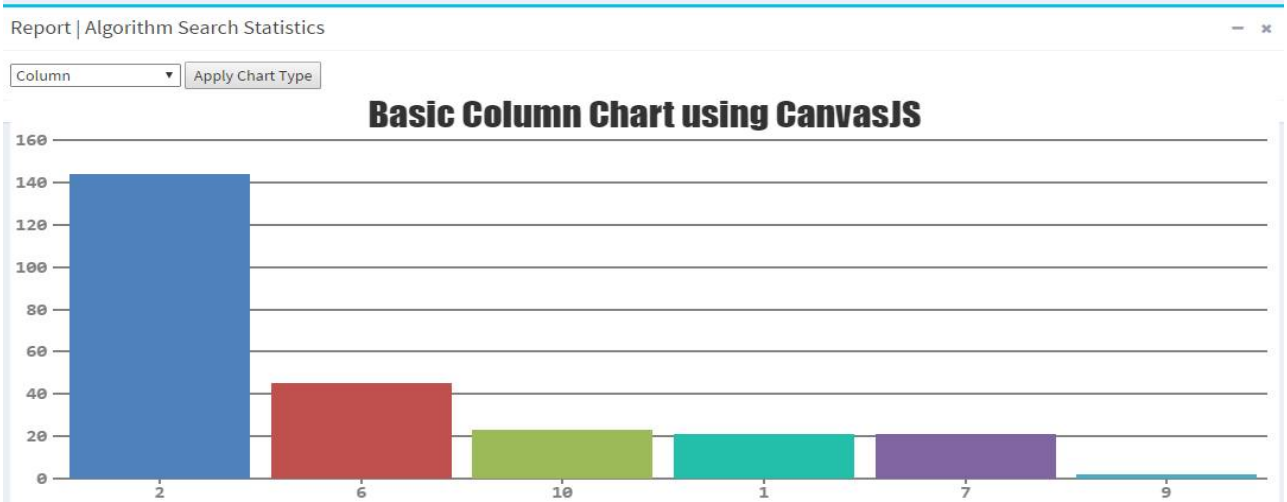
3) A most of sections are common sections such as Abstract, Introduction, Background, Conclusions, and References.

4.3.2 Using Document Sections for Linking Algorithm

Representations Assigning an AP to a section is easy, since it is part of the running text. Unlike APs, PCs are normally located separately from the running text; hence, they could appear outside the sections.. To get around this problem, a PC is mapped to the section that contains the largest number of reference sentences that refer to it. Once each algorithm representation is assigned to a section, algorithm representations which are mapped to the same sections are then linked.

Report Results:

Algorithm Search Statistics:



ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

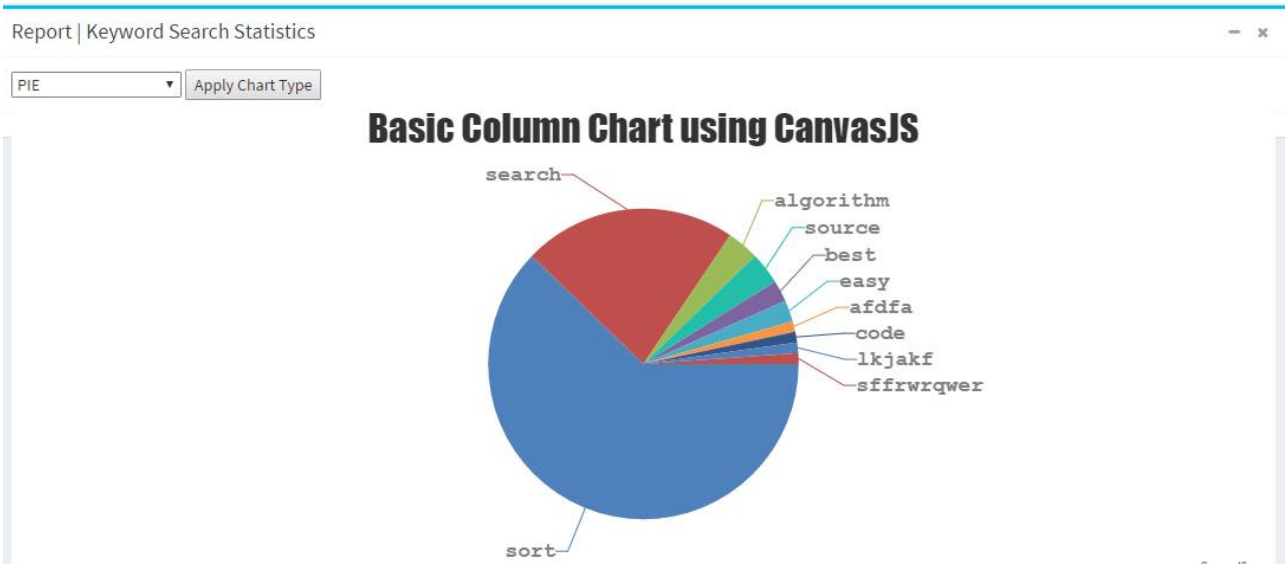
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Keyword Search Statistics:



Total Search Statistics:

Statistics | Total Search Statistics

Total no of Algorithms	10
Total no of Algorithms of type C	6
Total no of Algorithms of type Algorithm	3
Total no of Algorithms of type Pseudo	1
Total no of Words in Caption Text	96
Avg no of Words in Caption Text	27.272727272727%

V. CONCLUSION

Algorithms are very important in solving research problems. Scientificpublications host a tremendous amount of such high-quality algorithms developed by professional researchers. In digital libraries, being able to extract and catalogue these algorithms will introduce a number of exciting applications including algorithm searching, discovering, and analysing. Specifically, proposed system is a set of scalable machine learning based methods to detect algorithms in scholarly documents. The annotation methods for documents are extract textual metadata for pseudo-codes are discussed, and how algorithms are indexed and made searchable.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

VI. FUTURE WORK

Future work would be the semantic analysis of algorithms, their trends, and how algorithms influence each other over time. Such analyses would give rise to multiple applications that could improve algorithm search. We can add document files also; Robustness of the system will increase.

REFERENCES

[1] SuppawongTuarob, Member, IEEE, Sumit Bhatia, PrasenjitMitra, Senior Member,"AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data" IEEE TRANSACTIONS ON BIG DATA, VOL. 2, NO. 1, JANUARY-MARCH 2016.
[2] Sumit Bhatia, SuppawongTuarob, PrasenjitMitra and C. Lee Giles"An Algorithm Search Engine for Software Developers" The Pennsylvania StateUniversity Park, PA-16802, USA.
[3] Yves Petinot1, C. Lee Giles1,2,3, Vivek Bhatnagar2,3, Pradeep B. Teregowda1, Hui Han1"Enabling Interoperability For Autonomous.
[4] SuppawongTuarob, PrasenjitMitra and C. Lee Giles" Improving Algorithm Search Using the Algorithm Co-Citation Network"
[5] SuppawongTuarob, Sumit Bhatia, PrasenjitMitra, C. Lee Giles " Automatic Detection of Pseudocodes in Scholarly Documents Using Machine Learning".
[6] Kyle Williamsz, Jian Wuy, Sagnik Ray Choudhuryz, MadianKhabsay, C. Lee Gilesy"Scholarly Big Data Information Extraction and Integration in the CiteSeer_ Digital Library"Algorithm Procedures, Indexing, Mining and pseudo codes in big Data Algorithm Procedures, Indexing, Mining and pseudo codes in big Data